**Noname manuscript No.**
(will be inserted by the editor)

# Adaptive solution of linear systems of equations based on a posteriori error estimators

**A. Anciaux-Sedrakian · L. Grigori · Z. Jorti(✉) · J. Papež · S. Yousef**

Received: date / Accepted: date

**Abstract** In this paper we discuss a new adaptive approach for iterative solution of sparse linear systems arising from partial differential equations (PDE) with self-adjoint operators. The idea is to use the a posteriori estimated local distribution of the algebraic error in order to steer and guide the solve process in such way that the algebraic error is reduced more efficiently in the consecutive iterations. We first explain the motivation behind the proposed procedure and show that it can be equivalently formulated as constructing a special combination of preconditioner and initial guess for the original system. We present several numerical experiments in order to identify when the adaptive procedure can be of practical use.

**Keywords** Algebraic error · Adaptivity · Iterative solve · Preconditioning · Domain decomposition

**Mathematics Subject Classification (2010)** 65F08 · 65F10 · 65N22

## 1 Introduction

The seminal work as well as recent results on a posteriori error estimation allowed various adaptive concepts in numerical solution of partial differential equations (PDEs). For instance, an a posteriori local estimation of the

✉ Z. Jorti
IFP Energies nouvelles, 1-4 avenue de Bois-Préau, 92852 Rueil-Malmaison, France
INRIA Paris, Alpines, and Sorbonne Université, CNRS UMR 7598, Laboratoire Jacques-Louis Lions, Paris, France
Tel.: +33-642-755614
E-mail: zjorti@hotmail.fr

A. Anciaux-Sedrakian · S. Yousef
IFP Energies nouvelles, 1-4 avenue de Bois-Préau, 92852 Rueil-Malmaison, France

L. Grigori · J. Papež
INRIA Paris, Alpines, and Sorbonne Université, CNRS UMR 7598, Laboratoire Jacques-Louis Lions, Paris, France

discretization error (see, e.g., [2,20,4]) forms the basis for an adaptive mesh refinement. Such refinement can reduce the norm of the discretization error at a significantly lower cost in comparison to uniform mesh refinement, and typically results in a close-to-uniform spatial distribution of the error over the domain; see, e.g. [15]. These types of estimators, however, typically assume the exact solution of the associated algebraic system that is impossible to achieve in practice.

Inclusion of an inexact (approximate) algebraic solution into error estimators gave rise to *inexact* adaptive solution procedures, which, as a crucial ingredient, involve stopping criteria for iterative algebraic solvers; see, e.g., [1, Section 4]. The corresponding error estimators are typically decomposed into several parts that are identified with different components of the overall error, such as linearization, discretization and algebraic. The criteria in literature are based on well justified heuristics (see, e.g., [3,8]) and, recently in [16,17], also on mathematically rigorous proofs.

A common drawback of the above mentioned, rigorously justified estimators is their evaluation cost, which is typically (very) high with respect to the cost of an algebraic solver iteration. However, recent work [21] has resulted in the development of a posteriori estimates that can be easily coded, cheaply evaluated, and efficiently used in practical simulations providing a guaranteed control over different error components. They have confirmed that the computation of error estimators can be accessible even within non-academic contexts.

In this paper, we introduce a novel adaptive preconditioner for iteratively solving sparse linear systems arising from PDEs that modifies the iteration process according to the a posteriori estimated local distribution of the *algebraic error*. To best of our knowledge, there are yet no such procedures described in the literature. The paper therefore opens a discussion if adaptive approaches aiming at reducing the algebraic error in targeted parts of the domain are worth considering (at least in some cases) and how this aim can be achieved. In the paper we focus on self-adjoint PDE problems of second order only.

The adaptive procedure proposed in this paper can be briefly described as follows. In a given iteration step, based on the algebraic error distribution, a part of the solution domain and the associated algebraic degrees of freedom with high algebraic error are indicated. Then a block matrix splitting is introduced and used in a partitioned matrix procedure in order to yield, in the consecutive iterations, the residual vectors vanishing in the degrees of freedom indicated in the first step. We show that the proposed procedure corresponds to building, in a posteriori fashion based on information on the algebraic error at the above-mentioned iteration step, a particular combination of preconditioner and initial guess for following iterations. In addition, the sufficient and necessary conditions for attaining vanishing residuals are discussed.

The paper is organized as follows: Section 2 presents the model problem. In Section 3, we introduce a matrix splitting based on the distribution of algebraic errors, and discuss an approach related to minimization properties of preconditioned Conjugate Gradient method. In Section 4, we propose the

above mentioned adaptive procedure. In Section 5, we present and comment several numerical experiments before reaching a conclusion on when this procedure can be useful in accelerating the iterative solver. Finally, the conclusion overviews the work undertaken in this research and outlines directions for future study.

## 2 Model problem

This section introduces the model problem, and presents the key assumption that motivates the need for an adaptive solving procedure.
Let $\Omega \subset \mathbb{R}^d$, $1 \leq d \leq 3$ be a polytopal domain (open, bounded and connected set). We denote by $\overline{\Omega}$, $\Omega^o$, $\partial\Omega$ and $\mathcal{T}_h$ resp. the closure, interior, boundary and a matching simplicial mesh of $\Omega$. The extension of the results to nonmatching meshes is possible. We use the standard notation $\mathrm{L}^2(\Omega)$, $\mathrm{H}^1(\Omega)$ and $\mathrm{H}^1_0(\Omega)$ for the spaces of integrable functions, resp. integrable functions admitting weak derivations, and trace vanishing on $\partial\Omega$. For a vector $w$ of length $n \in \mathbb{N}$ and a subset $L \subset [\![1, n]\!]$, we denote by $w_L$ the restriction of $w$ to its components whose indexes belong to $L$.
Consider the problem that consists in seeking $\underline{u} : \Omega \to \mathbb{R}$ such that:

$$\begin{cases} -\nabla \cdot (\underline{\mathbf{K}} \nabla \underline{u}) = \underline{f} & \text{in } \Omega \\ \underline{u} = 0 & \text{on } \partial\Omega \end{cases} \tag{1}$$

where $\underline{f} : \Omega \to \mathbb{R}$ is a source term in $\mathrm{L}^2(\Omega)$, and $\underline{\mathbf{K}}$ is an uniformly bounded and positive definite diffusion tensor. For the sake of simplicity we assume that $\underline{f}$ and $\underline{\mathbf{K}}$ are piecewise constant with respect to the mesh $\mathcal{T}_h$. The weak form reads, find $\underline{u} \in V := \mathrm{H}^1_0(\Omega)$ such that

$$a(\underline{u}, \underline{v}) := (\underline{\mathbf{K}}^{\frac{1}{2}} \nabla \underline{u}, \underline{\mathbf{K}}^{\frac{1}{2}} \nabla \underline{v}) = (\underline{f}, \underline{v}) \qquad \forall \underline{v} \in V \tag{2}$$

where $a$ is a bilinear form. Associated to $\mathcal{T}_h$, let there be a discrete subspace $V_h \subset V$. The Galerkin solution $\underline{u}_h \in V_h$ satisfies

$$(\underline{\mathbf{K}}^{\frac{1}{2}} \nabla \underline{u}_h, \underline{\mathbf{K}}^{\frac{1}{2}} \nabla \underline{v}_h) = (\underline{f}, \underline{v}_h) \qquad \forall v_h \in V_h. \tag{3}$$

Considering a basis $(\varphi_l)_{1 \leq l \leq n}$ of $V_h$, this problem is equivalent to a system of linear algebraic equations:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b} \tag{4}$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a symmetric positive definite (SPD) matrix defined by $\mathbf{A}_{jk} = (\underline{\mathbf{K}}^{\frac{1}{2}} \nabla \varphi_k, \underline{\mathbf{K}}^{\frac{1}{2}} \nabla \varphi_j)$, $1 \leq j, k \leq n$, and $\mathbf{b} \in \mathbb{R}^n$ is the right hand side vector, $\mathbf{b}_j = (\underline{f}, \varphi_j)$. The continuous solution is then given by $\underline{u}_h = \sum_{l=1}^n x_l \varphi_l$. Let $\mathbf{x}^{(i)}$ be an approximate solution of (4) obtained after running $i$ iterations of an iterative solver, and $\underline{u}_h^{(i)} = \sum_{l=1}^n x_l^{(i)} \varphi_l$ the associated function from $V_h$. We

denote by $\mathbf{r}^{(i)}$ the corresponding algebraic residual vector $\mathbf{r}^{(i)} := \mathbf{b} - \mathbf{A} \cdot \mathbf{x}^{(i)}$. A relevant measure of the algebraic error is the energy norm

$$\|\underline{\mathbf{K}}^{\frac{1}{2}} \nabla(\underline{u}_h - \underline{u}_h^{(i)})\|_{\mathrm{L}^2(\Omega)} = \sqrt{a(\underline{u}_h - \underline{u}_h^{(i)}, \underline{u}_h - \underline{u}_h^{(i)})} = \|\mathbf{x} - \mathbf{x}^{(i)}\|_{\mathbf{A}} = \|\mathbf{r}^{(i)}\|_{\mathbf{A}^{-1}}, \tag{5}$$

where $\quad \forall \mathbf{y} \in \mathbb{R}^n, \quad \|\mathbf{y}\|_{\mathbf{A}^{-1}} := \mathbf{y}^{\mathrm{T}} \cdot \mathbf{A}^{-1} \cdot \mathbf{y}.$

The adaptive solution of linear systems proposed in this work is based on the fact that we can (tightly) estimate the local distribution of the error

$$\|\underline{\mathbf{K}}^{\frac{1}{2}} \nabla(\underline{u}_h - \underline{u}_h^{(i)})\|_{\mathrm{L}^2(K)}, \qquad \forall K \in \mathcal{T}_h, \tag{6}$$

where $K$ typically stands for the mesh elements ($\overline{\Omega} = \cup \overline{K}$). Based on this, we decompose the domain $\Omega$ into two disjoint parts $\Omega_1$ and $\Omega_2$:

$$\begin{cases} \overline{\Omega}_1 \cup \overline{\Omega}_2 & = \overline{\Omega} \\ \Omega_1^{\mathrm{o}} \cap \Omega_2^{\mathrm{o}} & = \emptyset \end{cases} \tag{7}$$

where $\Omega_1$ is the part with the high algebraic error:

$$\boxed{||\underline{\mathbf{K}}^{1/2} \nabla(\underline{u}_h - \underline{u}_h^{(i)})||^2_{\mathrm{L}^2(\Omega_1)} \gg ||\underline{\mathbf{K}}^{1/2} \nabla(\underline{u}_h - \underline{u}_h^{(i)})||^2_{\mathrm{L}^2(\Omega_2)}} \tag{8}$$

In fact, (8) is our main starting hypothesis. Figure 1 gives an illustrative example with $\Omega_1$ and $\Omega_2$ composed of a single element each.

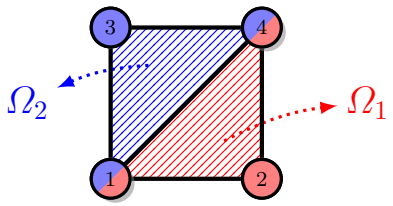



**Fig. 1** Simple example of the decomposition (7) with a $2 \times 2$ mesh grid.

## 3 Matrix decomposition and local error reduction

In this section, we introduce a sum splitting of the matrix $\mathbf{A}$ associated to the partitioning (7). Then, we discuss a first approach to locally reduce the local large algebraic errors. For that, we focus on the orthogonality properties that guarantee a decrease of the algebraic error norm in preconditioned conjugate gradient (PCG) solver before proposing a condition on the preconditioner that ensures that the algebraic error is locally decreased on the targeted subdomain.

### 3.1 Matrix decomposition: Sum splitting

According to the domain decomposition of (7) mentioned above, we denote by $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ the local stiffness matrices for the subdomains $\Omega_1$ and $\Omega_2$, respectively. While for the matrix $\mathbf{A}$ we have

$$\mathbf{A}_{jk} = (\underline{\mathbf{K}}^{\frac{1}{2}} \nabla \varphi_k, \underline{\mathbf{K}}^{\frac{1}{2}} \nabla \varphi_j), \qquad 1 \leq j,k \leq n \ ;$$

we define

$$\mathbf{A}^{(1)}_{jk} = (\underline{\mathbf{K}}^{\frac{1}{2}} \nabla \varphi_k, \underline{\mathbf{K}}^{\frac{1}{2}} \nabla \varphi_j)_{\Omega_1},\ 1 \leq j,k \leq n,\ \operatorname{supp} \varphi_k \cap \Omega_1 \neq \emptyset,\ \operatorname{supp} \varphi_j \cap \Omega_1 \neq \emptyset$$

$$\mathbf{A}^{(2)}_{jk} = (\underline{\mathbf{K}}^{\frac{1}{2}} \nabla \varphi_k, \underline{\mathbf{K}}^{\frac{1}{2}} \nabla \varphi_j)_{\Omega_2},\ 1 \leq j,k \leq n,\ \operatorname{supp} \varphi_k \cap \Omega_2 \neq \emptyset,\ \operatorname{supp} \varphi_j \cap \Omega_2 \neq \emptyset$$

For the ease of presentation we assume a convenient ordering such that the variables corresponding to the vertices of $\Omega_1$ are sorted first and those of $\Omega_2$ second. Then we can split the original operator, represented algebraically by the stiffness matrix $\mathbf{A}$, as follows

$$\mathbf{A} = \mathbf{A}^{(1)}_p + \mathbf{A}^{(2)}_p, \tag{9}$$

where $\mathbf{A}^{(1)}_p, \mathbf{A}^{(2)}_p$ are symmetric positive semidefinite (SPSD) with the following shapes:

$$\mathbf{A}^{(1)}_p = \left(\begin{array}{c|c} \mathbf{A}^{(1)} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array}\right) ; \qquad\qquad \mathbf{A}^{(2)}_p = \left(\begin{array}{c|c} \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{A}^{(2)} \end{array}\right) .$$

They are the extensions of the local stiffness matrices $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ to the whole domain. Then, we get the equivalent formulation of (8) in the matrix representation:

$$\boxed{(\mathbf{x} - \mathbf{x}^{(i)})^{\mathrm{T}} \cdot \mathbf{A}^{(1)}_p \cdot (\mathbf{x} - \mathbf{x}^{(i)}) \gg (\mathbf{x} - \mathbf{x}^{(i)})^{\mathrm{T}} \cdot \mathbf{A}^{(2)}_p \cdot (\mathbf{x} - \mathbf{x}^{(i)})}\ , \tag{10}$$

where $\mathbf{x}^{(i)}$ is the approximate solution at iteration $i$. Figure 2(a) illustrates how the global matrix $\mathbf{A}$ is built from $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$. The shaded part represents the common vertices between $\overline{\Omega}_1$ and $\overline{\Omega}_2$. It is the part of the matrix where the contributions from both subdomains are summed together.

### 3.2 Local error reduction with PCG

Let $\mathbf{x}^{(i)}$ be the approximate solution of (4) obtained at iteration $i$, with a PCG method using a preconditioner $\mathbf{M}$.

**Definition 1** Let $\mathbf{B}$ be a SPSD matrix and $(\mathbf{x}^{(j)})_j$ a sequence of vectors, $\mathbf{x}^{(j)} \underset{j}{\longrightarrow} \mathbf{x}$. We say that a $\mathbf{B}$-orthogonality property is satisfied when for every iteration $j$, we have

$$\langle \mathbf{B} \cdot (\mathbf{x} - \mathbf{x}^{(j+1)}), \mathbf{x}^{(j)} - \mathbf{x}^{(j+1)} \rangle = 0 \ . \tag{11}$$

**Lemma 1** *Let* $\mathbf{B}$ *be a SPSD matrix. The* $\mathbf{B}$*-orthogonality property satisfied for a sequence of vectors* $(\mathbf{x}^{(j)})_j$ *approximating the vector* $\mathbf{x}$*, ensures the decrease of the* $\mathbf{B}$*-seminorm of the algebraic error* $\langle \mathbf{B} \cdot (\mathbf{x} - \mathbf{x}^{(j)}), \mathbf{x} - \mathbf{x}^{(j)} \rangle$ *from any iteration* $j$ *to iteration* $j+1$*.*

*Proof* See [18, Section 5.2].

□

While iteratively solving (4), we could seek two orthogonalities in particular for the following reasons:

- **A-orthogonality**: It allows to minimize the global energy norm of the error according to Lemma 1.
- $\mathbf{A}_p^{(1)}$**-orthogonality**: It allows, according to Lemma 1, to reduce the dominant part of the global energy norm of the error as assumed in (10).

In our context, a primary goal is to reduce the $\mathbf{A}_p^{(1)}$-seminorm of the algebraic error, which is dominant according to the starting assumption (10). As stated in Lemma 1, the $\mathbf{A}_p^{(1)}$-orthogonality is a sufficient condition for the decrease of those quantities. For this reason, we now investigate means of ensuring the $\mathbf{A}_p^{(1)}$-orthogonality property. The $\mathbf{A}$-orthogonality is satisfied by definition thanks to the properties of the PCG method. Since we prefer to stay within the framework of PCG, there is no room in the choice of search directions. But the step size configuration constitutes a point for reflection, in the sense that there could exist some preconditioner $\mathbf{M}$ which yields particular step sizes such that the $\mathbf{A}_p^{(1)}$-orthogonality holds too.

**Lemma 2** *Consider a PCG iterative process [18, Algorithm 9.1,Chapter 9] to solve the linear system* (4)*, and denote by* $\mathbf{r}^{(j)}$ *and* $\mathbf{p}^{(j)}$ *the residual and descent direction respectively at iteration* $j$*. The* $\mathbf{A}$*-orthogonality property is guaranteed by the step size* $\alpha_j$*:*

$$\alpha_j = \frac{{\mathbf{r}^{(j)}}^{\mathrm{T}} \cdot \mathbf{M}^{-1} \cdot \mathbf{r}^{(j)}}{{\mathbf{p}^{(j)}}^{\mathrm{T}} \cdot \mathbf{A} \cdot \mathbf{p}^{(j)}} = \frac{{\mathbf{p}^{(j)}}^{\mathrm{T}} \cdot \mathbf{r}^{(j)}}{{\mathbf{p}^{(j)}}^{\mathrm{T}} \cdot \mathbf{A} \cdot \mathbf{p}^{(j)}} \ ; \tag{12}$$

*whereas the* $\mathbf{A}_p^{(1)}$*-orthogonality property can be guaranteed by the value taken by the step size* $\alpha_j$*:*

$$\alpha_j = \frac{{\mathbf{p}^{(j)}}^{\mathrm{T}} \cdot \mathbf{A}_p^{(1)} \cdot (\mathbf{x} - \mathbf{x}^{(j)})}{{\mathbf{p}^{(j)}}^{\mathrm{T}} \cdot \mathbf{A}_p^{(1)} \cdot \mathbf{p}^{(j)}} \ . \tag{13}$$

*Proof* The recurrence formulas of PCG give:

$$\begin{cases} \mathbf{x}^{(j+1)} = \mathbf{x}^{(j)} + \alpha_j \mathbf{p}^{(j)} \\ \mathbf{r}^{(j+1)} = \mathbf{r}^{(j)} - \alpha_j \mathbf{A} \cdot \mathbf{p}^{(j)} \end{cases}$$

Therefore,

$$\begin{aligned}(\mathbf{x}^{(j)}-\mathbf{x}^{(j+1)})^{\mathrm{T}}\cdot\mathbf{A}\cdot(\mathbf{x}-\mathbf{x}^{(j+1)})=0 &\iff (\alpha_j\mathbf{p}^{(j)})^{\mathrm{T}}\cdot\mathbf{r}^{(j+1)}=0\\ &\iff {\mathbf{p}^{(j)}}^{\mathrm{T}}\cdot(\mathbf{r}^{(j)}-\alpha_j\mathbf{A}\cdot\mathbf{p}^{(j)})=0\\ &\iff \alpha_j=\frac{{\mathbf{p}^{(j)}}^{\mathrm{T}}\cdot\mathbf{r}^{(i)}}{{\mathbf{p}^{(j)}}^{\mathrm{T}}\cdot\mathbf{A}\cdot\mathbf{p}^{(j)}}\end{aligned}$$

We can prove the $\mathbf{A}_p^{(1)}$-orthogonality with (13) analogously. The right part of the equality (12) can be demonstrated by induction from the recurrence formulas of PCG expressed above.

□

Lemma 2 gives step sizes that ensure $\mathbf{A}$-orthogonality and $\mathbf{A}_p^{(1)}$-orthogonality, respectively. Naturally, one can wonder when the two expressions (12) and (13) above match, i.e.:

$$\frac{{\mathbf{p}^{(j)}}^{\mathrm{T}}\cdot\mathbf{r}^{(j)}}{{\mathbf{p}^{(j)}}^{\mathrm{T}}\cdot\mathbf{A}\cdot\mathbf{p}^{(j)}}=\frac{{\mathbf{p}^{(j)}}^{\mathrm{T}}\cdot\mathbf{A}_p^{(1)}\cdot(\mathbf{x}-\mathbf{x}^{(j)})}{{\mathbf{p}^{(j)}}^{\mathrm{T}}\cdot\mathbf{A}_p^{(1)}\cdot\mathbf{p}^{(j)}} \tag{15}$$

The formula above cannot be used in practice because the term $\mathbf{A}_p^{(1)}\cdot(\mathbf{x}-\mathbf{x}^{(j)})$ cannot be computed as $\mathbf{x}$ is unknown. We outline here a special case when (15) holds: when $\mathbf{x}-\mathbf{x}^{(0)}$ is an eigenvector of $\mathbf{M}^{-1}\mathbf{A}$. Indeed, let $\lambda\in\mathbb{R}^*$ be the associated eigenvalue. We have:

$$\mathbf{p}^{(0)}:=\mathbf{M}^{-1}\cdot\mathbf{r}^{(0)}=\mathbf{M}^{-1}\mathbf{A}\cdot(\mathbf{x}-\mathbf{x}^{(0)})=\lambda(\mathbf{x}-\mathbf{x}^{(0)})\quad\text{and}\quad\mathbf{A}\cdot\mathbf{p}^{(0)}=\lambda\mathbf{r}^{(0)}$$

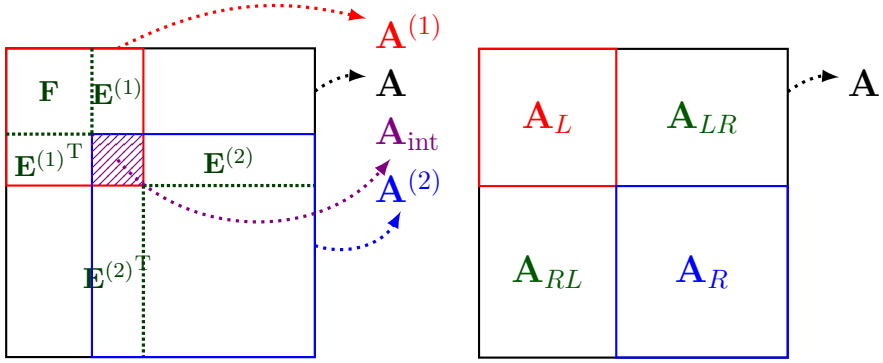
Hence,

$$\frac{{\mathbf{p}^{(0)}}^{\mathrm{T}}\cdot\mathbf{r}^{(0)}}{{\mathbf{p}^{(0)}}^{\mathrm{T}}\cdot\mathbf{A}\cdot\mathbf{p}^{(0)}}=\frac{1}{\lambda}\quad\text{and}\quad\frac{{\mathbf{p}^{(0)}}^{\mathrm{T}}\cdot\mathbf{A}_p^{(0)}\cdot(\mathbf{x}-\mathbf{x}^{(0)})}{{\mathbf{p}^{(0)}}^{\mathrm{T}}\cdot\mathbf{A}_p^{(0)}\cdot\mathbf{p}^{(0)}}=\frac{1}{\lambda}\ .$$

However, this assumption is too strong to be satisfied in practice for a PCG solver. In fact, it allows for the convergence in one iteration because:

$$\mathbf{M}^{-1}\mathbf{A}\cdot(\mathbf{x}-\mathbf{x}^{(0)})=\lambda(\mathbf{x}-\mathbf{x}^{(0)})\Longrightarrow\begin{cases}\mathbf{z}^{(0)}:=\mathbf{M}^{-1}\cdot\mathbf{r}^{(0)}=\lambda(\mathbf{x}-\mathbf{x}^{(0)});\\ \mathbf{p}^{(0)}:=\mathbf{z}^{(0)}=\lambda(\mathbf{x}-\mathbf{x}^{(0)});\\ \alpha^{(0)}:=\dfrac{{\mathbf{r}^{(0)}}^{\mathrm{T}}\cdot\mathbf{z}^{(0)}}{{\mathbf{p}^{(0)}}^{\mathrm{T}}\cdot\mathbf{A}\cdot\mathbf{p}^{(0)}}\\ \qquad=\dfrac{\lambda}{\lambda^2}\times\dfrac{(\mathbf{x}-\mathbf{x}^{(0)})^{\mathrm{T}}\mathbf{A}(\mathbf{x}-\mathbf{x}^{(0)})}{(\mathbf{x}-\mathbf{x}^{(0)})^{\mathrm{T}}\mathbf{A}(\mathbf{x}-\mathbf{x}^{(0)})}=\dfrac{1}{\lambda}\end{cases}$$
$$\Longrightarrow\mathbf{x}^{(1)}:=\mathbf{x}^{(0)}+\alpha^{(0)}\mathbf{p}^{(0)}=\mathbf{x}$$

This motivates seeking another procedure to ensure the local reduction of dominant errors.




(a) Splitting of matrix $\mathbf{A}$ with local stiffness matrices (b) Algebraic $2 \times 2$ block splitting of matrix $\mathbf{A}$

**Fig. 2** Splittings of matrix $\mathbf{A}$ with local stiffness matrices (left) and the associated algebraic $2 \times 2$ block splitting (right)

### 3.3 Matrix decomposition: Block partitioning

In this section, we derive a $2 \times 2$ block-partitioning of the matrix. This enables us to define a second approach based on appropriate initial guess and preconditioners to reduce the global error and make the residual nil in $\Omega_1$. In general, unless $\partial\Omega_1 \cap \partial\Omega \neq \emptyset$, the matrix $\mathbf{A}^{(1)}$ is singular. Since many common preconditioners and algebraic techniques (such as Cholesky factorization) are not suitable for a singular matrix, we replace the sum-splitting of the operator, as in (9), by a block partitioning of the matrix, such as

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_L & \mathbf{A}_{LR} \\ \mathbf{A}_{RL} & \mathbf{A}_R \end{pmatrix}. \tag{16}$$

Now if we decompose $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ as:

$$\mathbf{A}^{(1)} = \begin{pmatrix} \mathbf{F} & \mathbf{E}^{(1)} \\ {\mathbf{E}^{(1)}}^{\mathrm{T}} & \mathbf{A}^{(1)}_{\mathrm{int}} \end{pmatrix}; \qquad \mathbf{A}^{(2)} = \begin{pmatrix} \mathbf{A}^{(2)}_{\mathrm{int}} & \mathbf{E}^{(2)} \\ {\mathbf{E}^{(2)}}^{\mathrm{T}} & \mathbf{A}_R \end{pmatrix},$$

then the algebraic $2 \times 2$ block splitting of Figure 2(b) is built as follows:

$$\mathbf{A}_{LR} = \mathbf{A}_{RL}^{\mathrm{T}} = \begin{pmatrix} \mathbf{0} \\ \mathbf{E}^{(2)} \end{pmatrix}; \qquad \mathbf{A}_L = \mathbf{A}^{(1)} + \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^{(2)}_{\mathrm{int}} \end{pmatrix}, \tag{17}$$

where we denote:

- $L$: The set of nodes that belong to $\Omega_1$.
- $R$: The complementary of $L$.

Clearly, the matrix $\mathbf{A}_R$ does not contain any information about the common degrees of freedom, since the shaded part ($\mathbf{A}_{\mathrm{int}} = \mathbf{A}^{(1)}_{\mathrm{int}} + \mathbf{A}^{(2)}_{\mathrm{int}}$) is fully and exclusively integrated in $\mathbf{A}_L$. Note also that the matrices $\mathbf{A}_L$ and $\mathbf{A}_R$ are symmetric positive definite.

*Remark 1* The number of degrees of freedom of the overlapping part $\mathbf{A}_{\text{int}}$ depends on the algebraic error distribution. It may not be small with respect to the sizes of $\mathbf{A}_L$ and $\mathbf{A}_R$ respectively.

Splitting the vectors $\mathbf{b}$ and $\mathbf{x}$ according to the partitioning in (16) yields the vectors $\mathbf{b}_L$, $\mathbf{b}_R$, $\mathbf{x}_L$ and $\mathbf{x}_R$. Then, the corresponding block formulas for the solution $\mathbf{x}$ and the residual for $\mathbf{x}^{(i)}$ are:

$$\begin{bmatrix} \mathbf{b}_L \\ \mathbf{b}_R \end{bmatrix} = \begin{bmatrix} \mathbf{A}_L \cdot \mathbf{x}_L + \mathbf{A}_{LR} \cdot \mathbf{x}_R \\ \mathbf{A}_{RL} \cdot \mathbf{x}_L + \mathbf{A}_R \cdot \mathbf{x}_R \end{bmatrix} \tag{18}$$

$$\mathbf{b} - \mathbf{A} \cdot \mathbf{x}^{(i)} = \mathbf{A} \cdot (\mathbf{x} - \mathbf{x}^{(i)}) = \begin{bmatrix} \mathbf{A}_L \cdot (\mathbf{x} - \mathbf{x}^{(i)})_L + \mathbf{A}_{LR} \cdot (\mathbf{x} - \mathbf{x}^{(i)})_R \\ \mathbf{A}_{RL} \cdot (\mathbf{x} - \mathbf{x}^{(i)})_L + \mathbf{A}_R \cdot (\mathbf{x} - \mathbf{x}^{(i)})_R \end{bmatrix} \tag{19}$$

In the following, we present some properties of the submatrix $\mathbf{A}_L$, that allow to formulate an hypothesis for the algebraic errors that suits the $2 \times 2$ block splitting.

**Lemma 3 ($L$-Superiority wrt $\Omega_1$)** *Let* $\mathbf{w}$ *be an arbitrary vector. The following inequality holds:*

$$\mathbf{w}_L^{\mathrm{T}} \cdot \mathbf{A}_L \cdot \mathbf{w}_L \geq \mathbf{w}_L^{\mathrm{T}} \cdot \mathbf{A}^{(1)} \cdot \mathbf{w}_L \ .$$

*Proof* We know that $\mathbf{A}^{(2)}$ is a symmetric positive semi-definite (SPSD) matrix because it is the local stiffness matrix for subdomain $\Omega_2$. Since $\mathbf{A}_{\text{int}}^{(2)}$ is a principal submatrix of $\mathbf{A}^{(2)}$, it is SPSD as well. Therefore, the matrix $\mathbf{A}_L - \mathbf{A}^{(1)}$ from (17) is SPSD too and we have:

$$(\mathbf{w}_L^{\mathrm{T}} \cdot \mathbf{A}_L \cdot \mathbf{w}_L) - (\mathbf{w}_L^{\mathrm{T}} \cdot \mathbf{A}^{(1)} \cdot \mathbf{w}_L) = \mathbf{w}_L^{\mathrm{T}} \cdot (\mathbf{A}_L - \mathbf{A}^{(1)}) \cdot \mathbf{w}_L \geq 0 \ .$$

□

From this lemma, hypothesis (10) and the equality

$$(\mathbf{x} - \mathbf{x}^{(j)})^{\mathrm{T}} \cdot \mathbf{A}_p^{(1)} \cdot (\mathbf{x} - \mathbf{x}^{(j)}) = (\mathbf{x} - \mathbf{x}^{(j)})_L^{\mathrm{T}} \cdot \mathbf{A}^{(1)} \cdot (\mathbf{x} - \mathbf{x}^{(j)})_L \ ,$$

we can derive the subsequent corollary.

**Corollary 1 ($L$-Dominance)** *Let* $\mathbf{x}^{(j)}$ *be an arbitrary vector for which* (10) *is satisfied. Then the following inequalities hold:*

$$(\mathbf{x} - \mathbf{x}^{(j)})_L^{\mathrm{T}} \cdot \mathbf{A}_L \cdot (\mathbf{x} - \mathbf{x}^{(j)})_L \geq (\mathbf{x} - \mathbf{x}^{(j)})^{\mathrm{T}} \cdot \mathbf{A}_p^{(1)} \cdot (\mathbf{x} - \mathbf{x}^{(j)}) \gg (\mathbf{x} - \mathbf{x}^{(j)})^{\mathrm{T}} \cdot \mathbf{A}_p^{(2)} \cdot (\mathbf{x} - \mathbf{x}^{(j)}).$$

If we rewrite the contribution of each set to the energy norm of the error, we have for any approximate solution $\mathbf{x}^{(i)}$ of the initial system (4):

$$||\mathbf{x} - \mathbf{x}^{(i)}||_{\mathbf{A}}^2 = \langle \mathbf{A} \cdot (\mathbf{x} - \mathbf{x}^{(i)}), \mathbf{x} - \mathbf{x}^{(i)} \rangle = \underbrace{\langle \mathbf{b} - \mathbf{A} \cdot \mathbf{x}^{(i)}, \mathbf{x} - \mathbf{x}^{(i)} \rangle_L}_{:= L\text{-term}} + \underbrace{\langle \mathbf{b} - \mathbf{A} \cdot \mathbf{x}^{(i)}, \mathbf{x} - \mathbf{x}^{(i)} \rangle_R}_{:= R\text{-term}}, \tag{20}$$

where

$$\langle \mathbf{b} - \mathbf{A} \cdot \mathbf{x}^{(i)}, \mathbf{x} - \mathbf{x}^{(i)} \rangle_L = ||(\mathbf{x} - \mathbf{x}^{(i)})_L||_{\mathbf{A}_L}^2 + (\mathbf{x} - \mathbf{x}^{(i)})_L^{\mathrm{T}} \cdot \mathbf{A}_{LR} \cdot (\mathbf{x} - \mathbf{x}^{(i)})_R, \tag{21}$$

$$\langle \mathbf{b} - \mathbf{A} \cdot \mathbf{x}^{(i)}, \mathbf{x} - \mathbf{x}^{(i)} \rangle_R = ||(\mathbf{x} - \mathbf{x}^{(i)})_R||_{\mathbf{A}_R}^2 + (\mathbf{x} - \mathbf{x}^{(i)})_R^{\mathrm{T}} \cdot \mathbf{A}_{RL} \cdot (\mathbf{x} - \mathbf{x}^{(i)})_L \ .$$

Since $\mathbf{A}$ is symmetric, we have the equality of the coupling terms:

$$(\mathbf{x}-\mathbf{x}^{(i)})_R^{\mathrm{T}} \cdot \mathbf{A}_{RL} \cdot (\mathbf{x}-\mathbf{x}^{(i)})_L = (\mathbf{x}-\mathbf{x}^{(i)})_L^{\mathrm{T}} \cdot \mathbf{A}_{LR} \cdot (\mathbf{x}-\mathbf{x}^{(i)})_R \ .$$

We remind that PCG is known to minimize the global energy norm of the error $||\mathbf{x}-\mathbf{x}^{(i)}||_{\mathbf{A}}$, which is also equal to the algebraic error on the whole domain $\Omega$, following (5).
Yet, as expressed in Corollary 1, a concentrated algebraic error on a subdomain $\Omega_1$ implies that the $\mathbf{A}_L$-inner product of the error is dominant, and so will be the $L$-term, according to (21). This is why they should be reduced for an efficient decrease of the energy norm of the error. We recognize that reducing the $\mathbf{A}_L$-inner product is a rather delicate matter, because the vectors $(\mathbf{x} - \mathbf{x}^{(i)})_L$ and $\mathbf{A}_L \cdot (\mathbf{x} - \mathbf{x}^{(i)})_L$ are unknown. The alternative that we propose and we deem reasonable is to take into account a coupling term as well, in order to retrieve a partial residual $(\mathbf{b} - \mathbf{A} \cdot \mathbf{x}^{(i)})_L$ that is computable. Then, we can expect that in (20), the $L$-term is dominant in the global energy norms from the $i$-th iteration when hypothesis (10) holds, and we seek a process to efficiently decrease them during the preconditioned solve.

## 4 Adaptive preconditioner for PCG based on local error indicators

On the basis of the matrix decomposition described in 3.3, we introduce an adaptive preconditioning strategy enabling to reduce high local algebraic errors when solving the preconditioned linear system. The application of such a preconditioner starts after some few iterations that serve as an initialization phase, and is combined to a specific initial guess for the subsequent iterations of PCG solver. In the early stages of this study, our attention was clearly focused on substructuring methods that inspired us to apply to targeted error areas of the domain a similar treatment to the one foreseen for interface degrees of freedom in substructuring. The article [12] sets a reference framework for our study. It presents partitioned matrix methods along with Schur complement methods and establishes the equivalence between those two when PCG is used. It further suggests a more general form for the preconditioner where the local solves need not be carried out exactly.

### 4.1 Partitioned preconditioners suited for error reduction

As just explained above in Section 3.3, a good and affordable idea for ensuring the decay of local high algebraic errors seems to be to reduce the partial residual associated to the set of nodes $L$. A straightforward manner to bring that partial residual down to zero is by using a Schur complement reduction. If we consider the Schur complement matrix $\mathbf{S} := \mathbf{A}_R - \mathbf{A}_{RL}\mathbf{A}_L^{-1}\mathbf{A}_{LR}$, and the modified right hand side $\mathbf{g} := \mathbf{b}_R - \mathbf{A}_{RL}\mathbf{A}_L^{-1}\mathbf{b}_L$, we can start by iteratively solving the Schur complement system $\mathbf{S}\mathbf{x}_R = \mathbf{g}$ and then updating the other part of the solution by $\mathbf{x}_L := \mathbf{A}_L^{-1}\mathbf{b}_L - \mathbf{A}_L^{-1}\mathbf{A}_{LR} \cdot \mathbf{x}_R$. Note that it is this

update that guarantees the nil-residual on $L$-part. Nevertheless, $S$ is more likely to be dense even if the initial matrix $A$ is sparse. One way to avoid explicitly building $S$ is by taking advantage of a certain equivalence between the Schur complement procedure and a regular solve on the global system, with special initial guess and preconditioner. This equivalence is stated in the following theorem due to Eisenstat (see [12] and references therein).

**Theorem 1** *Using the same notation introduced above, let* $\mathbf{x}_S^{(k)}$ *be the* $k$*-th iterate of PCG solve of the system* $\mathbf{S}\cdot\mathbf{x}_S = \mathbf{g}$ *with initial guess* $\mathbf{x}_S^{(0)}$ *and preconditioner* $\mathbf{M}_S$*, and* $\mathbf{x}^{(k)}$ *be the* $k$*-th iterate of PCG solve of the system* $\mathbf{A}\cdot\mathbf{x} = \mathbf{b}$ *with initial guess* $\mathbf{x}^{(0)}$ *and preconditioner* $\mathbf{M}$ *such that:*

$$\mathbf{x}^{(0)} = \begin{bmatrix} \mathbf{A}_L^{-1}\cdot(\mathbf{b}_L - \mathbf{A}_{LR}\cdot\mathbf{x}_S^{(0)}) \\ \mathbf{x}_S^{(0)} \end{bmatrix} \;;\; \mathbf{M} = \begin{pmatrix} \mathbf{A}_L & \mathbf{A}_{LR} \\ \mathbf{A}_{RL} & \mathbf{M}_S + \mathbf{A}_{RL}\mathbf{A}_L^{-1}\mathbf{A}_{LR} \end{pmatrix}. \tag{22}$$

*Then there holds, at each iteration* $k$*,*

$$\mathbf{x}^{(k)} = \begin{bmatrix} \mathbf{A}_L^{-1}\cdot(\mathbf{b}_L - \mathbf{A}_{LR}\cdot\mathbf{x}_S^{(k)}) \\ \mathbf{x}_S^{(k)} \end{bmatrix}.$$

*Proof* We give here an alternative proof to the one given in [12, Theorem 2.1 (i)] as we demonstrate by induction the results stated. We keep the notation used in [18, Algorithm 9.1,Chapter 9] for the PCG algorithm for solving (4) and we use an $S$ subscript for the formulas associated to the system $\mathbf{S}\cdot\mathbf{x}_S = \mathbf{g}$.
Note that the inverse of $\mathbf{M}$ can be expressed as:

$$\mathbf{M}^{-1} = \begin{pmatrix} \mathbf{A}_L^{-1} + \mathbf{A}_L^{-1}\mathbf{A}_{LR}\mathbf{M}_S^{-1}\mathbf{A}_{RL}\mathbf{A}_L^{-1} & -\mathbf{A}_L^{-1}\mathbf{A}_{LR}\mathbf{M}_S^{-1} \\ -\mathbf{M}_S^{-1}\mathbf{A}_{RL}\mathbf{A}_L^{-1} & \mathbf{M}_S^{-1} \end{pmatrix}.$$

Next, we proceed by induction on $k \in \mathbb{N}$ to prove that:

$$\mathbf{x}^{(k)} = \begin{bmatrix} \mathbf{A}_L^{-1}\cdot(\mathbf{b}_L - \mathbf{A}_{LR}\cdot\mathbf{x}_S^{(k)}) \\ \mathbf{x}_S^{(k)} \end{bmatrix};\; \mathbf{r}^{(k)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{r}_S^{(k)} \end{bmatrix};\; \mathbf{z}^{(k)} = \begin{bmatrix} -\mathbf{A}_L^{-1}\mathbf{A}_{LR}\cdot\mathbf{z}_S^{(k)} \\ \mathbf{z}_S^{(k)} \end{bmatrix}$$
$$\mathbf{p}^{(k)} = \begin{bmatrix} -\mathbf{A}_L^{-1}\mathbf{A}_{LR}\cdot\mathbf{p}_S^{(k)} \\ \mathbf{p}_S^{(k)} \end{bmatrix} \;;\; (\mathbf{r}^{(k)})^{\mathrm{T}}\mathbf{z}^{(k)} = (\mathbf{r}_S^{(k)})^{\mathrm{T}}\mathbf{z}_S^{(k)} \;;\; \alpha^{(k)} = \alpha_S^{(k)}\;.$$

For $k = 0$: the first equality is satisfied by definition for $\mathbf{x}^{(0)}$. For the other equalities we have:

$$\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\cdot\mathbf{x}^{(0)} = \begin{bmatrix}\mathbf{b}_L\\ \mathbf{b}_R\end{bmatrix} - \begin{bmatrix}\mathbf{b}_L - \mathbf{A}_{LR}\cdot\mathbf{x}_S^{(0)} + \mathbf{A}_{LR}\cdot\mathbf{x}_S^{(0)}\\ \mathbf{A}_{RL}\mathbf{A}_L^{-1}\cdot(\mathbf{b}_L - \mathbf{A}_{LR}\cdot\mathbf{x}_S^{(0)}) + \mathbf{A}_R\cdot\mathbf{x}_S^{(0)}\end{bmatrix}$$
$$= \begin{bmatrix}\mathbf{0}\\ \mathbf{g} - \mathbf{S}\cdot\mathbf{x}_S^{(0)}\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\ \mathbf{r}_S^{(0)}\end{bmatrix}$$

$$\mathbf{z}^{(0)} = \mathbf{M}^{-1}\cdot\mathbf{r}^{(0)} = \mathbf{M}^{-1}\cdot\begin{bmatrix}\mathbf{0}\\ \mathbf{r}_S^{(0)}\end{bmatrix} = \begin{bmatrix}-\mathbf{A}_L^{-1}\mathbf{A}_{LR}\mathbf{M}_S^{-1}\cdot\mathbf{r}_S^{(0)}\\ \mathbf{M}_S^{-1}\cdot\mathbf{r}_S^{(0)}\end{bmatrix} = \begin{bmatrix}-\mathbf{A}_L^{-1}\mathbf{A}_{LR}\cdot\mathbf{z}_S^{(0)}\\ \mathbf{z}_S^{(0)}\end{bmatrix}$$

$$\mathbf{p}^{(0)} = \mathbf{z}^{(0)} = \begin{bmatrix}-\mathbf{A}_L^{-1}\mathbf{A}_{LR}\cdot\mathbf{z}_S^{(0)}\\ \mathbf{z}_S^{(0)}\end{bmatrix} = \begin{bmatrix}-\mathbf{A}_L^{-1}\mathbf{A}_{LR}\cdot\mathbf{p}_S^{(0)}\\ \mathbf{p}_S^{(0)}\end{bmatrix};$$

$$\mathbf{A}\cdot\mathbf{p}^{(0)} = \begin{bmatrix}\mathbf{0}\\ -\mathbf{A}_{RL}\mathbf{A}_L^{-1}\mathbf{A}_{LR}\cdot\mathbf{p}_S^{(0)} + \mathbf{A}_R\cdot\mathbf{p}_S^{(0)}\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\ \mathbf{S}\cdot\mathbf{p}_S^{(0)}\end{bmatrix}.$$

Then, $\quad \mathbf{p}^{(0)^{\mathrm{T}}}\mathbf{A}\cdot\mathbf{p}^{(0)} = \mathbf{p}_S^{(0)^{\mathrm{T}}}\mathbf{S}\cdot\mathbf{p}_S^{(0)}$.

$$\mathbf{r}^{(0)^{\mathrm{T}}}\mathbf{z}^{(0)} = \mathbf{r}_S^{(0)^{\mathrm{T}}}\mathbf{z}_S^{(0)};\quad \text{therefore}\quad \alpha^{(0)} = \frac{\mathbf{r}^{(0)^{\mathrm{T}}}\mathbf{z}^{(0)}}{\mathbf{p}^{(0)^{\mathrm{T}}}\mathbf{A}\cdot\mathbf{p}^{(0)}} = \frac{\mathbf{r}_S^{(0)^{\mathrm{T}}}\mathbf{z}_S^{(0)}}{\mathbf{p}_S^{(0)^{\mathrm{T}}}\mathbf{S}\cdot\mathbf{p}_S^{(0)}} = \alpha_S^{(0)}.$$

Let $k \in \mathbb{N}$, we assume the equalities above are true for $k$, we then have:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{p}^{(k)} = \begin{bmatrix}\mathbf{A}_L^{-1}\cdot(\mathbf{b}_L - \mathbf{A}_{LR}\cdot\mathbf{x}_S^{(k)})\\ \mathbf{x}_S^{(k)}\end{bmatrix} + \alpha_S^{(k)}\begin{bmatrix}-\mathbf{A}_L^{-1}\mathbf{A}_{LR}\cdot\mathbf{p}_S^{(k)}\\ \mathbf{p}_S^{(k)}\end{bmatrix}$$
$$= \begin{bmatrix}\mathbf{A}_L^{-1}\cdot(\mathbf{b}_L - \mathbf{A}_{LR}\cdot(\mathbf{x}_S^{(k)} + \alpha_S^{(k)}\mathbf{p}_S^{(k)}))\\ \mathbf{x}_S^{(k)} + \alpha_S^{(k)}\mathbf{p}_S^{(k)}\end{bmatrix} = \begin{bmatrix}\mathbf{A}_L^{-1}\cdot(\mathbf{b}_L - \mathbf{A}_{LR}\cdot\mathbf{x}_S^{(k+1)})\\ \mathbf{x}_S^{(k+1)}\end{bmatrix}$$

$$\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha^{(k)}\mathbf{A}\cdot\mathbf{p}^{(k)} = \begin{bmatrix}\mathbf{0}\\ \mathbf{r}_S^{(k)}\end{bmatrix} - \alpha_S^{(k)}\begin{bmatrix}\mathbf{0}\\ -\mathbf{A}_{RL}\mathbf{A}_L^{-1}\mathbf{A}_{LR}\cdot\mathbf{p}_S^{(k)} + \mathbf{A}_R\cdot\mathbf{p}_S^{(k)}\end{bmatrix}$$
$$= \begin{bmatrix}\mathbf{0}\\ \mathbf{r}_S^{(k)} - \alpha_S^{(k)}\mathbf{S}\cdot\mathbf{p}_S^{(k)}\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\ \mathbf{r}_S^{(k+1)}\end{bmatrix};$$

$$\mathbf{z}^{(k+1)} = \mathbf{M}^{-1}\cdot\mathbf{r}^{(k+1)} = \begin{bmatrix}-\mathbf{A}_L^{-1}\mathbf{A}_{LR}\mathbf{M}_S^{-1}\cdot\mathbf{r}_S^{(k+1)}\\ \mathbf{M}_S^{-1}\cdot\mathbf{r}_S^{(k+1)}\end{bmatrix} = \begin{bmatrix}-\mathbf{A}_L^{-1}\mathbf{A}_{LR}\mathbf{z}_S^{(k+1)}\\ \mathbf{z}_S^{(k+1)}\end{bmatrix}.$$

As a consequence, we have:

$$(\mathbf{r}^{(k+1)})^{\mathrm{T}}\mathbf{z}^{(k+1)} = (\mathbf{r}_S^{(k+1)})^{\mathrm{T}}\mathbf{z}_S^{(k+1)}$$

Combining this latter equality with the one stemming from the previous step, we obtain

$$\beta^{(k)} = \frac{(\mathbf{r}^{(k+1)})^{\mathrm{T}}\mathbf{z}^{(k+1)}}{(\mathbf{r}^{(k)})^{\mathrm{T}}\mathbf{z}^{(k)}} = \frac{(\mathbf{r}_S^{(k+1)})^{\mathrm{T}}\mathbf{z}_S^{(k+1)}}{(\mathbf{r}_S^{(k)})^{\mathrm{T}}\mathbf{z}_S^{(k)}} = \beta_S^{(k)}.$$

Thus,

$$\mathbf{p}^{(k+1)} = \begin{bmatrix} -\mathbf{A}_L^{-1}\mathbf{A}_{LR}\cdot(\mathbf{z}_S^{(k+1)} + \beta_S^{(k)}\mathbf{p}_S^{(k)}) \\ \mathbf{z}_S^{(k+1)} + \beta_S^{(k)}\mathbf{p}_S^{(k)} \end{bmatrix} = \begin{bmatrix} -\mathbf{A}_L^{-1}\mathbf{A}_{LR}\cdot\mathbf{p}_S^{(k+1)} \\ \mathbf{p}_S^{(k+1)} \end{bmatrix};$$

$$\mathbf{A}\cdot\mathbf{p}^{(k+1)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{S}\cdot\mathbf{p}_S^{(k+1)} \end{bmatrix};$$

Finally $\alpha^{(k+1)} = \dfrac{{\mathbf{r}^{(k+1)}}^{\mathrm{T}}\mathbf{z}^{(k+1)}}{{\mathbf{p}^{(k+1)}}^{\mathrm{T}}\mathbf{A}\cdot\mathbf{p}^{(k+1)}} = \dfrac{{\mathbf{r}_S^{(k+1)}}^{\mathrm{T}}\mathbf{z}_S^{(k+1)}}{{\mathbf{p}_S^{(k+1)}}^{\mathrm{T}}\mathbf{S}\cdot\mathbf{p}_S^{(k+1)}} = \alpha_S^{(k+1)}.$$

□

In the remainder of this section, we generalize Theorem 1 to provide sufficient and necessary conditions on the initial guess and the preconditioner to obtain a nil residual on $L$ at each iteration.

**Theorem 2 (Sufficient condition for nil residual on $L$-part)** *We denote* $\mathbf{W}$ *the Cholesky factor of* $\mathbf{M}$*:* $\mathbf{M} = \mathbf{W}\mathbf{W}^{\mathrm{T}}$*,* $n_L$ *and* $n_R$ *designate the sizes of the diagonal blocks* $\mathbf{A}_L$ *and* $\mathbf{A}_R$ *respectively. Let* $\mathbf{x}_R^{(0)}$ *be an arbitrary vector of length* $n_R$ *and* $\mathbf{W}_1$*,* $\mathbf{W}_2$ *two invertible matrices of sizes* $n_L$ *and* $n_R$ *respectively. Let the linear system* (4) *be solved by a PCG solver with a preconditioner* $\mathbf{M} = \mathbf{W}\mathbf{W}^{\mathrm{T}}$ *and an initial guess* $\mathbf{x}^{(0)}$ *such that:*

$$\mathbf{x}^{(0)} = \begin{bmatrix} \mathbf{A}_L^{-1}\cdot(\mathbf{b}_L - \mathbf{A}_{LR}\cdot\mathbf{x}_R^{(0)}) \\ \mathbf{x}_R^{(0)} \end{bmatrix}, \quad \mathbf{W} = \begin{pmatrix} \mathbf{W}_1 & \mathbf{0} \\ \mathbf{A}_{RL}\mathbf{A}_L^{-1}\mathbf{W}_1 & \mathbf{W}_2 \end{pmatrix}; \tag{23}$$

*then* $(\mathbf{b} - \mathbf{A}\cdot\mathbf{x}^{(k)})_L = \mathbf{0}$ *at each iteration* $k$ *of PCG.*

*Proof* With

$$\mathbf{x}^{(0)} = \begin{bmatrix} \mathbf{A}_L^{-1}\cdot(\mathbf{b}_L - \mathbf{A}_{LR}\cdot\mathbf{x}_R^{(0)}) \\ \mathbf{x}_R^{(0)} \end{bmatrix},$$

there holds:

$$\mathbf{r}^{(0)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{b}_R - \mathbf{A}_{RL}\mathbf{A}_L^{-1}\cdot\mathbf{b}_L + (\mathbf{A}_{RL}\mathbf{A}_L^{-1}\mathbf{A}_{LR} - \mathbf{A}_R)\cdot\mathbf{x}_R^{(0)} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{g} - \mathbf{S}\cdot\mathbf{x}_R^{(0)} \end{bmatrix}.$$

Therefore

$$\mathbf{W}^{-1}\cdot\mathbf{r}^{(0)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{W}_2^{-1}\cdot(\mathbf{g} - \mathbf{S}\cdot\mathbf{x}_R^{(0)}) \end{bmatrix}.$$

Besides, for each iteration $k$ there exists a polynomial $q_k$ of degree $k$ such that:

$$\mathbf{W}^{-1}\cdot\mathbf{r}^{(k)} = q_k(\mathbf{W}^{-1}\mathbf{A}\mathbf{W}^{-\mathrm{T}})\mathbf{W}^{-1}\cdot\mathbf{r}^{(0)} \tag{24}$$

The definition of preconditioner $\mathbf{W}$ in (23) yields:

$$\mathbf{W}^{-1}\mathbf{A}\mathbf{W}^{-\mathrm{T}} = \begin{pmatrix} \mathbf{W}_1^{-1}\mathbf{A}_L\mathbf{W}_1^{-\mathrm{T}} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_2^{-1}\mathbf{S}\mathbf{W}_2^{-\mathrm{T}} \end{pmatrix} \tag{25}$$

Then

$$q_k(\mathbf{W}^{-1}\mathbf{A}\mathbf{W}^{-\mathrm{T}}) = \begin{pmatrix} q_k(\mathbf{W}_1^{-1}\mathbf{A}_L\mathbf{W}_1^{-\mathrm{T}}) & \mathbf{0} \\ \mathbf{0} & q_k(\mathbf{W}_2^{-1}\mathbf{S}\mathbf{W}_2^{-\mathrm{T}}) \end{pmatrix}$$

Consequently, from (24) we deduce that:

$$\mathbf{r}^{(k)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{W}_2 q_k(\mathbf{W}_2^{-1}\mathbf{S}\mathbf{W}_2^{-\mathrm{T}})\mathbf{W}_2^{-1}\cdot(\mathbf{g} - \mathbf{S}\cdot\mathbf{x}_R^{(0)}) \end{bmatrix}$$

□

*Remark 2* When the sufficient condition above is satisfied, the preconditioner $\mathbf{M}$ has the following shape:

$$\begin{aligned}\mathbf{M} &= \begin{pmatrix} \mathbf{W}_1 & \mathbf{0} \\ \mathbf{A}_{RL}\mathbf{A}_L^{-1}\mathbf{W}_1 & \mathbf{W}_2 \end{pmatrix}\begin{pmatrix} \mathbf{W}_1^{\mathrm{T}} & \mathbf{W}_1^{\mathrm{T}}\mathbf{A}_L^{-\mathrm{T}}\mathbf{A}_{LR} \\ \mathbf{0} & \mathbf{W}_2^{\mathrm{T}} \end{pmatrix} \\ &= \left(\begin{array}{c|c} \mathbf{W}_1\mathbf{W}_1^{\mathrm{T}} & \mathbf{W}_1\mathbf{W}_1^{\mathrm{T}}\mathbf{A}_L^{-\mathrm{T}}\mathbf{A}_{LR} \\ \hline \mathbf{A}_{RL}\mathbf{A}_L^{-1}\mathbf{W}_1\mathbf{W}_1^{\mathrm{T}} & \mathbf{W}_2\mathbf{W}_2^{\mathrm{T}} + \mathbf{A}_{RL}\mathbf{A}_L^{-1}\mathbf{W}_1\mathbf{W}_1^{\mathrm{T}}\mathbf{A}_L^{-\mathrm{T}}\mathbf{A}_{LR} \end{array}\right)\end{aligned}$$

If we denote the two SPD matrices $\mathbf{M}_1 := \mathbf{W}_1\mathbf{W}_1^{\mathrm{T}}$ and $\mathbf{M}_2 := \mathbf{W}_2\mathbf{W}_2^{\mathrm{T}}$ then:

$$\mathbf{M} = \left(\begin{array}{c|c} \mathbf{M}_1 & \mathbf{M}_1\mathbf{A}_L^{-\mathrm{T}}\mathbf{A}_{LR} \\ \hline \mathbf{A}_{RL}\mathbf{A}_L^{-1}\mathbf{M}_1 & \mathbf{M}_2 + \mathbf{A}_{RL}\mathbf{A}_L^{-1}\mathbf{M}_1\mathbf{A}_L^{-\mathrm{T}}\mathbf{A}_{LR} \end{array}\right);$$

which is a generalization of the preconditioner defined in (22) of Theorem 1.

When the conditions of Theorem 2 are fulfilled, we can state that in addition to the $\mathbf{A}$-orthogonality property, the residual vanishes on $L$ at each iteration. The question that arises next is to know if the preconditioner defined in Theorem 2 is the only one that has this particular property or if there exist others.

**Theorem 3 (Necessary condition for nil residual on $L$-part)** *Let $\mathbf{M}$ be a preconditioner of $\mathbf{A}$ such that the solve of (4) by PCG yields a vanishing residual on $L$ at each iteration:*

$$\mathbf{r}_L^{(k)} = \mathbf{b}_L - \mathbf{A}_L\cdot\mathbf{x}_L^{(k)} - \mathbf{A}_{LR}\cdot\mathbf{x}_L^{(k)} = 0\ , \qquad \forall k \leq \tilde{k} \quad ;$$

*where $\tilde{k}$ is the iteration when $\mathbf{x}^{(\tilde{k})} = \mathbf{x}$. Then we have:*

$$\dim(\mathrm{Ker}((\mathbf{A}\mathbf{M}^{-1})_{LR})) \geq \tilde{k} - 1\ . \tag{26}$$

*Proof* Still with the notation used in [18, Algorithm 9.1,Chapter 9] for the PCG algorithm, the successive residuals satisfy a two-term recurrence:

$$\mathbf{r}^{(i+1)} = -\alpha^{(i)}\mathbf{A}\mathbf{M}^{-1}\cdot\mathbf{r}^{(i)} + (1 + \frac{\alpha^{(i)}\beta^{(i-1)}}{\alpha^{(i-1)}})\mathbf{r}^{(i)} - \frac{\alpha^{(i)}\beta^{(i-1)}}{\alpha^{(i-1)}}\mathbf{r}^{(i-1)}, \forall i \in \mathbb{N}\backslash\{0\} \tag{27}$$

Let $i \in [\![1, \tilde{k}]\!]$ be arbitrary. A vanishing residual on $L$ implies that

$$\mathbf{r}_L^{(i+1)} = \mathbf{r}_L^{(i)} = \mathbf{r}_L^{(i-1)} = 0\ ,$$

then due to (27), we get

$$(\mathbf{A}\mathbf{M}^{-1} \cdot \mathbf{r}^{(i)})_L = 0 \ ;$$

which using the $2 \times 2$ block splitting for $\mathbf{A}\mathbf{M}^{-1}$ as in (16) and the fact that $\mathbf{r}_L^{(i)} = 0$ gives

$$(\mathbf{A}\mathbf{M}^{-1})_{LR} \cdot \mathbf{r}_R^{(i)} = 0 \tag{28}$$

We know that the vectors $(\mathbf{x} - \mathbf{x}^{(0)}, \ldots, \mathbf{x} - \mathbf{x}^{(k)})$ are linearly independent for every $k < \tilde{k}$. Because $\mathbf{A}$ is nonsingular, $(\mathbf{r}^{(0)} = \mathbf{A} \cdot (\mathbf{x} - \mathbf{x}^{(0)}), \ldots, \mathbf{r}^{(k)} = \mathbf{A} \cdot (\mathbf{x} - \mathbf{x}^{(k)}))$ are linearly independent as well. Since $\mathbf{r}_L^{(j)} = 0$ for $j = 0, \ldots, k$, we deduce that $(\mathbf{r}_R^{(0)}, \ldots, \mathbf{r}_R^{(k)})$ are also linearly independent for $k < \tilde{k}$.
According to (28), this implies that $\dim(\mathrm{Ker}((\mathbf{A}\mathbf{M}^{-1})_{LR})) \geq k \ , \forall k < \tilde{k}$
i.e. $\dim(\mathrm{Ker}((\mathbf{A}\mathbf{M}^{-1})_{LR})) \geq \tilde{k} - 1$.

□

Theorems 2 and 3 provide respectively sufficient and necessary conditions on the preconditioner to obtain a nil residual on $L$ at each iteration of PCG. There is a special case when these conditions match each other and the proposed preconditioner is unique. Indeed, for $\tilde{k} > n_R$, we have $\dim(\mathrm{Ker}((\mathbf{A}\mathbf{M}^{-1})_{LR})) \geq n_R$ and thus according to the rank-nullity theorem:

$$\mathrm{rank}((\mathbf{A}\mathbf{M}^{-1})_{LR}) = n_R - \dim(\mathrm{Ker}((\mathbf{A}\mathbf{M}^{-1})_{LR})) \leq 0.$$

Which means that $(\mathbf{A}\mathbf{M}^{-1})_{LR} = 0$. This latter equality is equivalent to the block diagonal shape (25) in the proof of Theorem 2.

### 4.2 Condition number improvement

In this subsection, we discuss the outcome of using the partitioned preconditioner of Theorem 1 with respect to a $2 \times 2$ block diagonal preconditioner in terms of a possible reduction of the condition number of the preconditioned operator. Considering $\mathbf{M}$ (resp. $\mathbf{M}_S$) the SPD preconditioner of $\mathbf{A}$ (resp. $\mathbf{S}$), we denote the following condition numbers:

$$\mathcal{K}(\mathbf{A}, \mathbf{M}) := \mathcal{K}(\mathbf{M}^{-\frac{1}{2}}\mathbf{A}\mathbf{M}^{-\frac{1}{2}}) \ ;$$
$$\mathcal{K}(\mathbf{S}, \mathbf{M}_S) := \mathcal{K}(\mathbf{M}_S^{-\frac{1}{2}}\mathbf{S}\mathbf{M}_S^{-\frac{1}{2}}) \ .$$

According to [14, Theorem 4.2.], for every block-diagonal preconditioner $\widetilde{\mathbf{M}}$ of $\mathbf{A}$ which has the shape:

$$\widetilde{\mathbf{M}} = \left(\begin{array}{c|c} \mathbf{A}_L & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{M}_R \end{array}\right) \tag{29}$$

such that $\mathbf{M}_R$ is an arbitrary preconditioner associated with the second diagonal block $\mathbf{A}_R$, we have:

$$\mathcal{K}(\mathbf{S}, \mathbf{M}_R) \leq \mathcal{K}(\mathbf{A}, \widetilde{\mathbf{M}})$$

Therefore, if we choose $\mathbf{M}_S = \mathbf{M}_R$, we still have:

$$\mathcal{K}(\mathbf{S}, \mathbf{M}_S) \leq \mathcal{K}(\mathbf{A}, \widetilde{\mathbf{M}}) \tag{30}$$

This means that the procedure discussed above yields a preconditioned system with the condition number equal or lower than that of the original system preconditioned by the block diagonal matrix $\widetilde{\mathbf{M}}$. Later, in Section 5, we will show the effect of the convergence improvement on the number of iterations in numerical experiments.

### 4.3 Context of use

Let us suggest the context of the solution process for practical use.

❶ A PCG solver with a given preconditioner $\overline{\mathbf{M}}$ is run to solve (4).
❷ At an iteration step $j_0$, an intermediate solution $\mathbf{x}^{(j_0)}$ is computed to get an estimated algebraic error distribution.
↪ Local algebraic error $\eta_{\text{alg},K}^{(j_0)}$ is evaluated over each mesh elements $K \in \mathcal{T}_h$.
❸ This allows for marking the elements with largest errors and extracting their associated node indices.
↪ This step yields subdomain $\Omega_1$ of (7) and (8), and $L$-subset.
❹ Proceed with permuting the system to obtain a $L/R$ splitting as in (16).
❺ Perform an exact Cholesky factorization on the $L$-block, build the adaptive preconditioner and use $\mathbf{x}^{(j_0)}$ to compute a special initial guess.
↪ $\mathbf{W}_1$(such that $\mathbf{W}_1\mathbf{W}_1^{\mathrm{T}}$=$\mathbf{A}_L$), $\mathbf{x}^{(0)}$ and $\mathbf{M}$ as defined in (22) with $\mathbf{x}_S^{(0)} := \mathbf{x}_R^{(j_0)}$.
❻ Run PCG on the permuted system with preconditioner $\mathbf{M}$ and the newly computed starting guess $\mathbf{x}^{(0)}$.

*Remark 3* Concerning step n°5 above, for computing $\mathbf{M}$ in (22), the $\mathbf{M}_S$ matrix should ideally approximate the Schur complement matrix $\mathbf{S}$. Though there already exists a wide range of Schur complement methods [19,5,13], we suggest to recycle the preconditioner $\overline{\mathbf{M}}$ by extracting its $R$-block and using it as $\mathbf{M}_S$. This allows to save the time required to construct $\mathbf{M}$.

For the sake of comparison, we denote by process 2 the solve procedure described above, and by process 1 the solve with preconditioner $\overline{\mathbf{M}}$ on the initial system pursued to the end. We can say that the cost should differ between the two processes. In process 2, more effort is put into building the preconditioner. The additional cost compared to process 1 is the one related to the Cholesky factorization of $\mathbf{A}_L$. However, the cost of the solve is hopefully diminished due to the convergence improvement with $\mathbf{M}$ over $\overline{\mathbf{M}}$.

## 5 Numerical results

In this section, we choose different 2D elliptic problems as they usually serve as test cases for PDE resolution algorithms. In order to assess the efficiency

of the adaptive procedure, the first numerical results presented in Sections 5.2 and 5.3 rely on using the exact algebraic error. In Sections 5.4 and 5.5 we also present numerical results that are based on local error estimates, and hence of practical interest. The following numerical experiments are based on Matlab with PDE toolbox in order to create a mesh and solve the considered PDE on a domain $\Omega$.
We consider a triangular mesh. If the exact solution is known, the mesh is adaptively refined according to the distribution of the discretization error during an initialization step before running tests. Otherwise, the mesh is Delaunay and is generated by Matlab initmesh command with the maximum element size specified by the parameter $H_{\max}$. We will call such mesh "uniform" in the sequel. Once the main linear system is defined, we run few iterations of the linear solver (20 iterations of PCG) to get a starting distribution of the algebraic error on all the elements of the mesh. From these quantities, we distinguish the $L$ and $R$ subdomains. Multiple strategies are conceivable for selecting the elements that will form our $L$-subdomain. We will define some of them in Section 5.1. Next, we solve the linear system by using the procedure described in Section 4.3. Finally, we compare the evolution of the global energy norm and the $L$-norm of the error (that we define in Section 5.1) during the iterative solve for process 1 and process 2.

### 5.1 Some strategies for initiating the adaptive procedure

In order to build the $L$-subdomain, we start by sorting the mesh elements according to the algebraic errors per element. Then in order to select elements that will compose $L$, we need a certain threshold. One option is to take a ratio on the number of elements. It consists in setting a certain percentage "*perc*" on the total number of elements, and gathering the first *perc* % elements where the algebraic error is the largest. However, due to the complex distribution of errors over elements for some test cases, we believe it would be a good idea to adjust the way we choose elements that form $\Omega_1$. Henceforth, we apply the so-called Dörfler criterion [7]. It aims at finding the minimal set $\mathcal{E}_1$ within the set of all elements $\mathcal{E}$ such that for some parameter $\Theta \in ]0,1[$ :

$$L\text{-norm}^2 := \sum_{K\in\mathcal{E}_1} (\eta_{\text{alg},K})^2 \geq \Theta \left( \sum_{K\in\mathcal{E}} (\eta_{\text{alg},K})^2 \right) , \tag{31}$$

where $\eta_{\text{alg},K}$ denotes the algebraic error over the element $K$, and the term "$L$-norm" in this article will refer, somewhat imprecisely, to the portion of error captured in $\Omega_1 = \bigcup_{K\in\mathcal{E}_1} K$. Note that it is actually the $\mathbf{A}_p^{(1)}$-seminorm of the error:

$$L\text{-norm}^2 = \langle \mathbf{A}_p^{(1)} \cdot (\mathbf{x} - \mathbf{x}^{(j_0)}), \mathbf{x} - \mathbf{x}^{(j_0)} \rangle .$$

In our framework, we believe that (31) better reflects Hypothesis (8). Indeed, the advantage over the previous marking strategy is that we are selecting

elements that concentrate a certain percentage of the error instead of directly choosing a percentage of the total number of elements.

*Remark 4* Whenever the discretization error is known, we could exploit it as an indicator of the right iteration to simulate the starting algebraic error distribution. Indeed, the evaluation of errors would initiate as soon as the iterative solve reaches an iteration $j_0$ such that:

$$\eta_{\text{alg}}^{(j_0)} \leq \gamma \, \eta_{\text{disc}}^{(j_0)} \, ,$$

where $\eta_{\text{alg}}^{(j_0)}$ and $\eta_{\text{disc}}^{(j_0)}$ are the total algebraic and discretization errors respectively, $\gamma > 0$ is a scalar parameter. We expect that this approach would give reliable information on the error distribution since inequalities of this shape have been used as stopping criterion in many adaptive algorithms in the literature; see, e.g., [6,8] and the references given there.

For the numerical experiments presented in this section, we decide to consider:

- a fixed value (20) for $j_0$,
- a Block-Jacobi preconditioner composed of 50 blocks for $\overline{\mathbf{M}}$,
- a stopping threshold value of $10^{-6}$ for the euclidean norm of the residual.

In the following, whenever the initial algebraic error distribution is plotted, thick horizontal lines labeled with $\theta_1$ and $\theta_2$ on the color bar indicates the extent of the errors' range covered with the Dörfler rates $\Theta_1$ and $\Theta_2$ considered, i.e. all elements represented in color shades above the corresponding thresholds $\theta_1$ and $\theta_2$ respectively form $\Omega_1$. As far as the convergence curves are concerned, the blue one corresponds to process 1 while process 2 is represented in red and black. Detailed information about test configuration and results are given in Table 1.

### 5.2 Poisson's equation

First, we focus exclusively on Poisson equations of the form

$$-\Delta \underline{u} = -\frac{\partial^2 \underline{u}}{\partial x^2} - \frac{\partial^2 \underline{u}}{\partial y^2} = \underline{f}(x,y) \quad \text{in} \quad \Omega \tag{32}$$

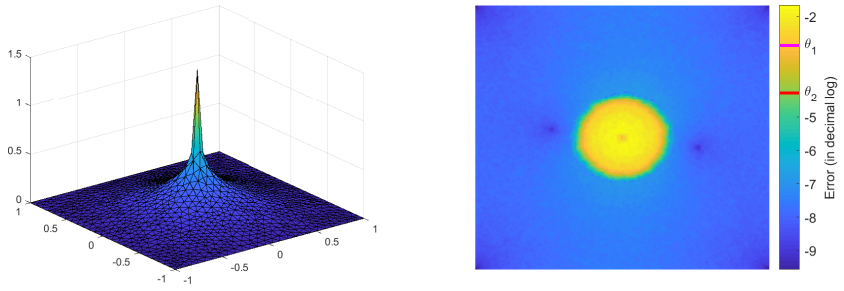
with homogeneous Dirichlet boundary condition

$$\underline{u} = \underline{u}_0 \quad \text{on} \quad \partial\Omega. \tag{33}$$

This is a particular case of problem (1) with the diffusion tensor equal to identity. For our test cases, we consider two classic examples with given smooth solutions $\underline{u}$:

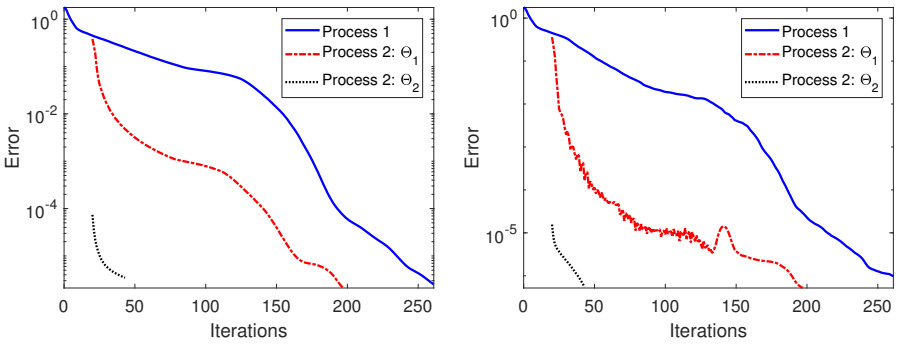
$$\underline{u}^{(1)} = (x+1) \times (x-1) \times (y+1) \times (y-1) \times \exp(-\alpha \times (x^2+y^2)) \; ; \tag{34}$$

$$\underline{u}^{(2)} = (x+1) \times (x-1) \times (y+1) \times (y-1) \times (\exp(-\alpha \times ((x+0.5)^2 + (y+0.5)^2)) - \exp(-\beta \times ((x-0.5)^2 + (y-0.5)^2))) \; ; \tag{35}$$




(a) Galerkin solution $u_h^{(1)}$ (b) Algebraic errors after 20 iterations

**Fig. 3** Galerkin solution and initial algebraic error distribution for test case n°1

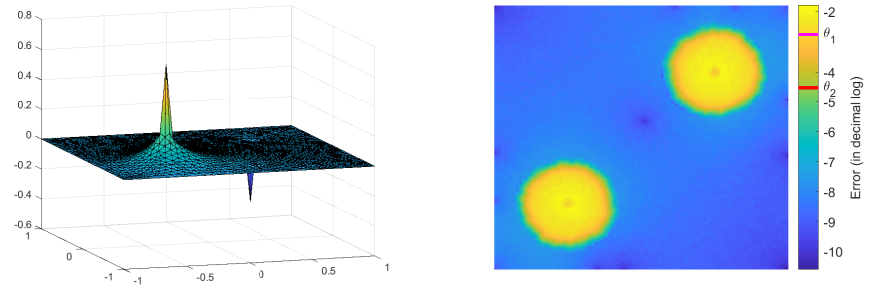



(a) Energy norm of the algebraic error (b) $L$-norm of the algebraic error

**Fig. 4** Error evolution for test case n°1

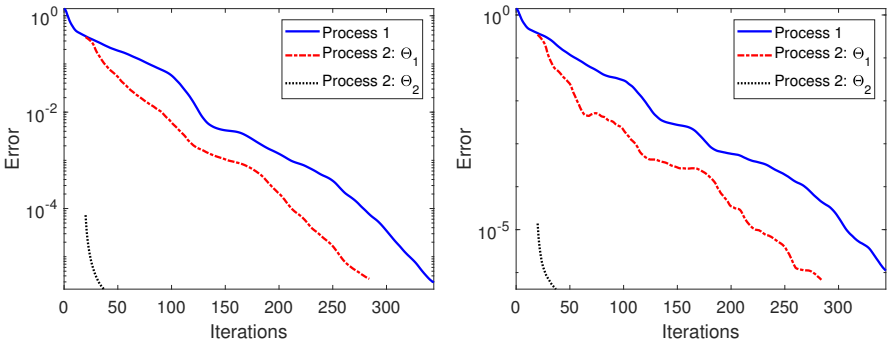
with $\alpha = 4\,000$ and $\beta = 3\,000$.

For the test case with $\underline{u}^{(1)}$ defined in (34), we consider an adaptively refined mesh with maximum edge size $H_{\max} = 0.02$ and the total number of elements equals to 36 370. After discretization, the size of the matrix $\mathbf{A}$ is $17\,986 \times 17\,986$. The Galerkin solution and the initial algebraic error distribution (after $j_0 = 20$ iterations) over the domain $\Omega$ are shown in Figure 3, while the global energy norm and the $L$-norm of the error within iterations are shown in Figure 4 for two different values of the parameter $\Theta$ in the Dörfler criterion. With a Dörfler rate $\Theta_1 := 0.98$, $\Omega_1$ does not take into account all the area where important errors are observed. We notice in Figure 4 the decrease of the global energy norm of the error and that of the algebraic error on $L$-marked elements is observed on both processes but more markedly for the process 2. Now, when we increase the Dörfler rate to $\Theta_2 := 0.99$ to cover almost all the high errors' region, we notice the rapid decrease of the global energy norm of the error and that of the algebraic error on $L$-marked elements with process 2 (black




(a) Galerkin solution $u_h^{(2)}$ (b) Algebraic errors after 20 iterations

**Fig. 5** Galerkin solution and initial algebraic error distribution for test case n°2

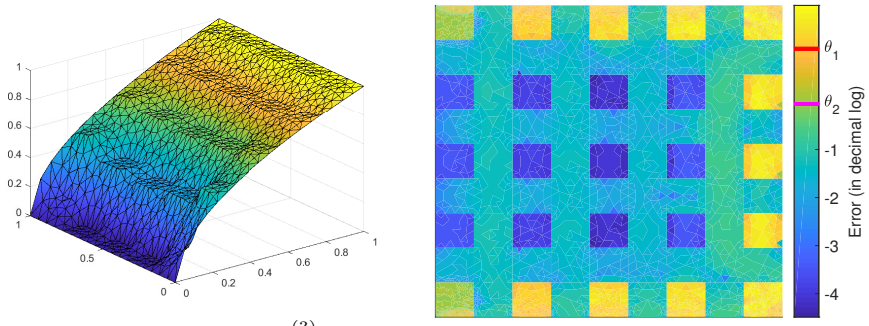



(a) Energy norm of the algebraic error (b) $L$-norm of the algebraic error

**Fig. 6** Error evolution for test case n°2

curve). The convergence is faster and there is a major 6x speedup in terms of iterations.

For the test case with the exact solution $\underline{u}^{(2)}$ defined by (35), we consider an adaptively refined mesh, with maximum edge size $H_{\max} = 0.02$; then the total number of elements is 38 384. After discretization, the size of the matrix $\mathbf{A}$ is 18 993 $\times$ 18 993. The Galerkin solution and the initial algebraic error distribution over the domain $\Omega$ are plotted in Figure 5, while the global energy norm and the $L$-norm of the error within iterations are depicted in Figure 6. In this latter figure, we point out that the error reduction in process 2 with $\Theta_1 := 0.98$ is better than in process 1. It becomes even more efficient with $\Theta_2 := 0.99$ as it yields a convergence after 37 iterations only. That represents an important 19x speedup over PCG. These results outline the potential of the adaptive method even for the case when the errors are localized in separate zones. We provide more experiments replacing the distribution of the algebraic error with the local error estimates in Sections 5.4 and 5.5.




(a) Galerkin solution $u_h^{(3)}$ (b) Algebraic errors after 20 iterations

**Fig. 7** Galerkin solution and initial algebraic error distribution for test case n°3

### 5.3 Diffusion equation with inhomogeneous coefficient

In this section, we tackle diffusion problems with inhomogeneous diffusion tensor of the form

$$-\nabla \cdot (\underline{\mathbf{K}} \nabla \underline{u}) = \underline{f}(x, y) \quad \text{in} \quad \Omega \tag{36}$$

with Dirichlet boundary condition

$$\underline{u} = \underline{u}_0 \quad \text{on} \quad \partial\Omega \tag{37}$$

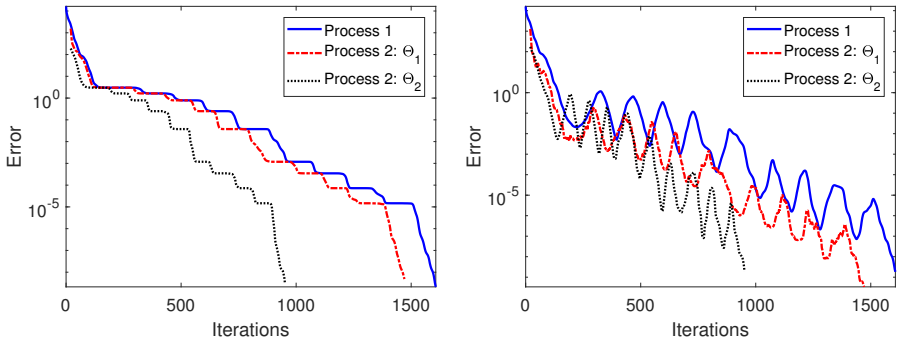
This time, we define the right hand side $\underline{f}$ as the constant function taking the value 1 on $\Omega$.
The Dirichlet boundary condition is prescribed on $\partial\Omega$ by the function:

$$\underline{u}_0(x, y) = \sqrt{x}; \tag{38}$$

The diffusivity taken here is a highly heterogeneous function of $\Omega$. In the sequel, we will consider two configurations of the diffusivity. In both cases, the diffusion tensor is defined as a multiple of the identity matrix: $\underline{\mathbf{K}} = c\mathbf{I}$; and the multiplication factor $c$ varies through the domain $\Omega$. In the first test, the diffusivity is defined as in [11, Section 5]:

$$c^{(3)}(x, y) = \begin{cases} 10^5(\lfloor 9x \rfloor + 1) & \text{if} \quad \lfloor (9x) \rfloor \equiv 0 \pmod 2 \text{ and } \lfloor (9y) \rfloor \equiv 0 \pmod 2, \\ 1 & \text{otherwise.} \end{cases}$$

We consider a uniform mesh with a maximum edge size $H_{\max} = 0.03$ and 30 257 mesh elements. After discretization, the size of the matrix $\mathbf{A}$ is $14\,690 \times 14\,690$. The Galerkin solution and the initial algebraic error distribution over the domain $\Omega$ are shown in Figure 7, while the global energy norm and the $L$-norm of the error within iterations are plotted in Figure 8. We observe that curves of process 2 are almost below the curve of process 1 for the $L$-norm in Figure 8. We also highlight that by going from $\Theta_1 = 0.95$ to $\Theta_2 = 0.99$ to take into account more error zones, we increase the size of $\mathbf{A}_L$ by a factor of three,




(a) Energy norm of the algebraic error (b) $L$-norm of the algebraic error

**Fig. 8** Error evolution for test case n°3

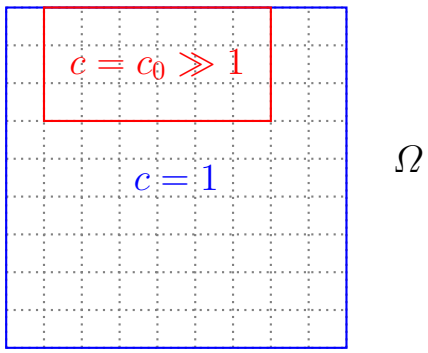


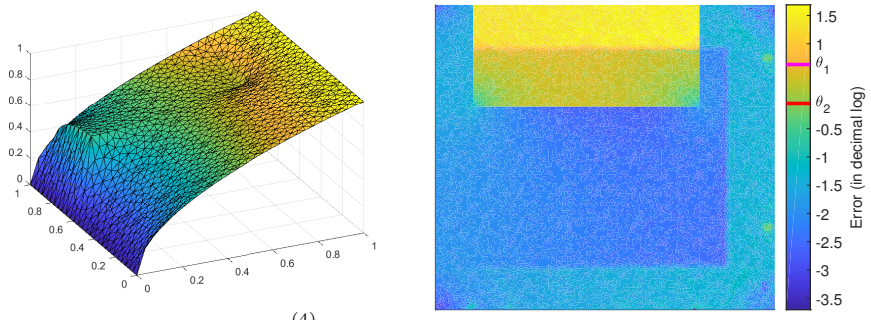

**Fig. 9** Configuration of the inhomogeneous diffusivity in test case n°4

but the reduction in number of iterations with respect to PCG was five times larger ($8\% \rightarrow 41\%$).

For the next test case, we consider the second diffusivity shown in Figure 9. The formula of the corresponding diffusivity function is:

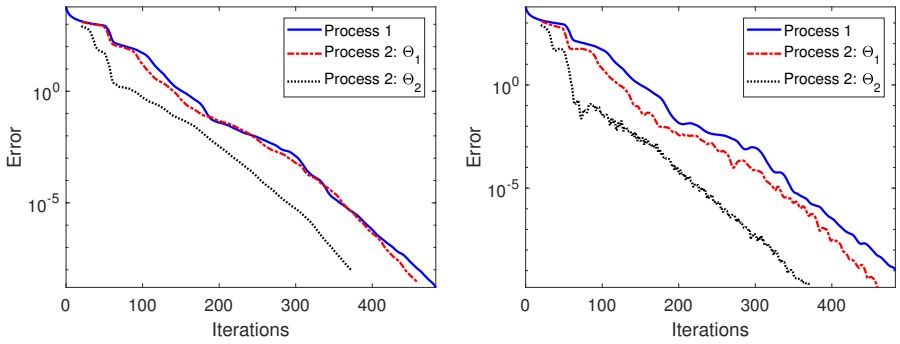
$$c^{(4)}(x,y) = \begin{cases} c_0 := 9 \times 10^5 & \text{if} \quad \lfloor (9x) \rfloor \in [1,7] \text{ and } \lfloor (9y) \rfloor \in [6,9] \\ 1 & \text{otherwise} \end{cases}$$

This time, with a maximum edge size $H_{\max} = 0.01$, we obtain a uniform mesh of 32 544 elements. The size of the matrix $\mathbf{A}$ is $16\,057 \times 16\,057$. The Galerkin solution and the initial algebraic error distribution over the domain $\Omega$ are depicted in Figure 10, while the global energy norm and the $L$-norm of the error during iterations are plotted in Figure 11. We start with a value $\Theta_1 = 0.81$ that ensures that the size of the submatrix $\mathbf{A}_L$ is about one tenth of the size of the global matrix $\mathbf{A}$. While the blue and red curves of Figure 11 are very close to each other for the global energy norm, we observe that the red one is always below the blue one when it comes to the $L$-norm.




(a) Galerkin solution $u_h^{(4)}$ (b) Algebraic errors after 20 iterations

**Fig. 10** Galerkin solution and initial algebraic error distribution for test case n°4




(a) Energy norm of the algebraic error (b) $L$-norm of the algebraic error

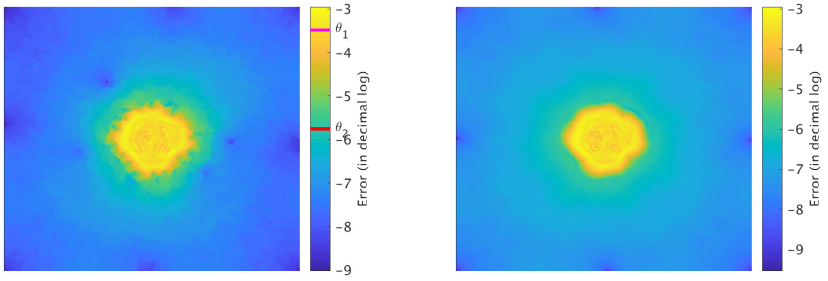
**Fig. 11** Error evolution for test case n°4

For this test case, we see that the high algebraic errors are concentrated on a rectangle. With the Dörfler rate $\Theta_1$, we capture only an upper band of that rectangle. As a consequence, the curves associated to process 2 will not show any substantial improvement over those of process 1 with that setting.
On the other hand, increasing the value of the Dörfler rate and considering $\Theta_2 = 0.99$ ensures that the $L$-subdomain covers almost all the rectangle with large algebraic errors. We obtain a submatrix $\mathbf{A}_L$ whose size is about 20% of the size of the global matrix $\mathbf{A}$. Now, we see in Figure 11 that the curves of process 1 (blue) and process 2 (black) are distinct.

In almost all of the test cases presented above (except test case n°3), we have voluntarily restricted ourselves to taking subdomains $\Omega_1$ whose size $n_L$ does not exceed 25% of $n$ the size of global matrix in order to avoid configurations where process 2 is too costly with respect to process 1. However, it is important to bear in mind that the ideal size of the $L$-subdomain cannot be always limited under a certain threshold and is linked to the distribution of the algebraic errors for the problem considered.

**Table 1** Test configuration and number of iterations for standard and adaptive processes with different values of the Dörfler rate $\Theta$.

| Configuration | | | Iterations | |
|---|---|---|---|---|
| Test case | $\Theta$ (as %) | $n_L$ in % of $n$ | $it_{\text{st}}$ | $it_{\text{ada}}$ |
| 1 | 98 | 12.02 | 241 | 177 |
| | 99.99 | 13.74 | 241 | 23 |
| 2 | 98 | 18.64 | 324 | 264 |
| | 99.99 | 25.17 | 324 | 17 |
| 3 | 95 | 23.38 | 1587 | 1451 |
| | 99.93 | 69.98 | 1587 | 931 |
| 4 | 81 | 10 | 463 | 440 |
| | 99.99 | 21.68 | 463 | 353 |



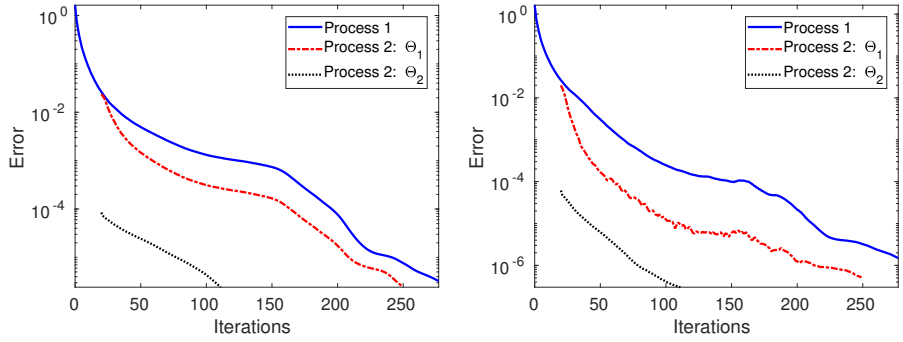

(a) Algebraic a posteriori error estimates after 20 iterations

(b) Algebraic errors after 20 iterations

**Fig. 12** Initial distribution and a posteriori estimation of algebraic error for test case n°1 on mesh $\mathcal{M}^{(1)}$

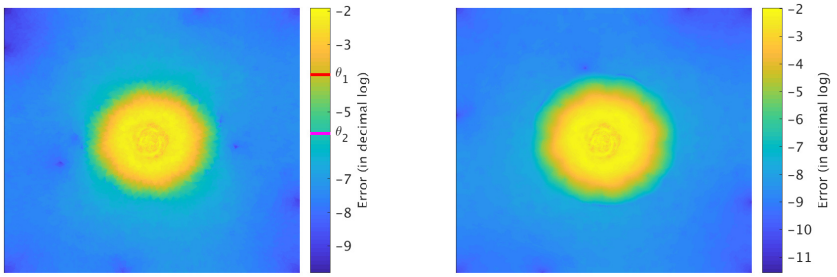
## 5.4 Numerical results for Poisson's equation with a posteriori error estimates

In this section, we reproduce the numerical experiments of Section 5.2 using a posteriori algebraic error estimates of [16], without evaluating the exact algebraic error. We consider two uniform meshes $\mathcal{M}^{(1)}$ and $\mathcal{M}^{(2)}$ with maximum edge sizes $H_{\max} = 0.1$ and $H_{\max} = 0.05$ respectively. The total number of elements is equal to 87 552 and 354 304 respectively. After discretization, the matrices have dimensions $43\,457 \times 43\,457$ and $176\,513 \times 176\,513$ respectively. For the first test case, the initial distribution and a posteriori estimation of algebraic error (after $j_0 = 20$ iterations) over the domain $\Omega$ are shown in Figure 12 for the first mesh $\mathcal{M}^{(1)}$ and in Figure 14 for the second mesh $\mathcal{M}^{(2)}$. The global energy norm and the $L$-norm of the error within iterations are shown in Figures 13 (for $\mathcal{M}^{(1)}$) and 15 (for $\mathcal{M}^{(2)}$) with two different values of the parameter $\Theta$ in the Dörfler criterion. With the first Dörfler rate $\Theta_1$, $\Omega_1$ does not include the entire subdomain where important errors are observed. We notice in Figures 13 and 15 the decrease of the global energy norm of the



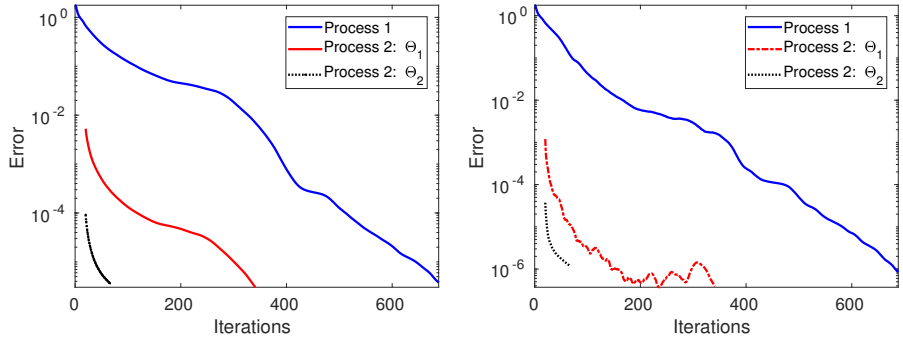

(a) Energy norm of the algebraic error (b) $L$-norm of the algebraic error

**Fig. 13** Error evolution for test case n°1 on mesh $\mathcal{M}^{(1)}$ using a posteriori error estimates



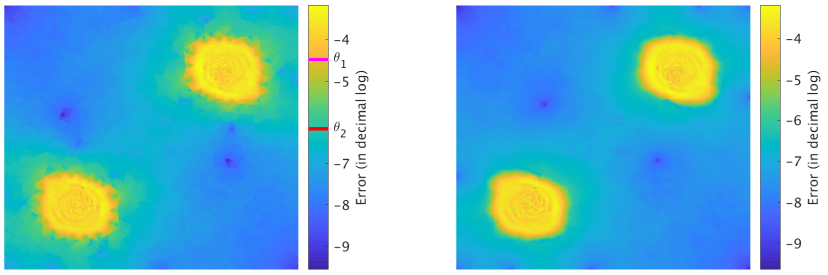

(a) Algebraic a posteriori error estimates after 20 iterations (b) Algebraic errors after 20 iterations

**Fig. 14** Initial distribution and a posteriori estimation of algebraic error for test case n°1 on mesh $\mathcal{M}^{(2)}$



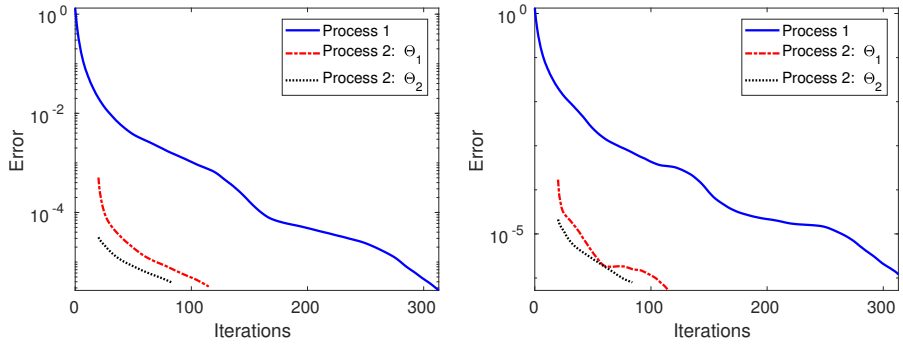

(a) Energy norm of the algebraic error (b) $L$-norm of the algebraic error

**Fig. 15** Error evolution for test case n°1 on mesh $\mathcal{M}^{(2)}$ using a posteriori error estimates




(a) Algebraic a posteriori error estimates after 20 iterations (b) Algebraic errors after 20 iterations

**Fig. 16** Initial distribution and a posteriori estimation of algebraic error for test case n°2 on mesh $\mathcal{M}^{(1)}$

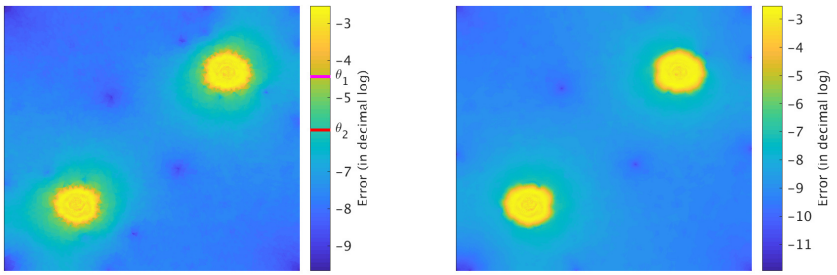



(a) Energy norm of the algebraic error (b) $L$-norm of the algebraic error

**Fig. 17** Error evolution for test case n°2 on mesh $\mathcal{M}^{(1)}$ using a posteriori error estimates

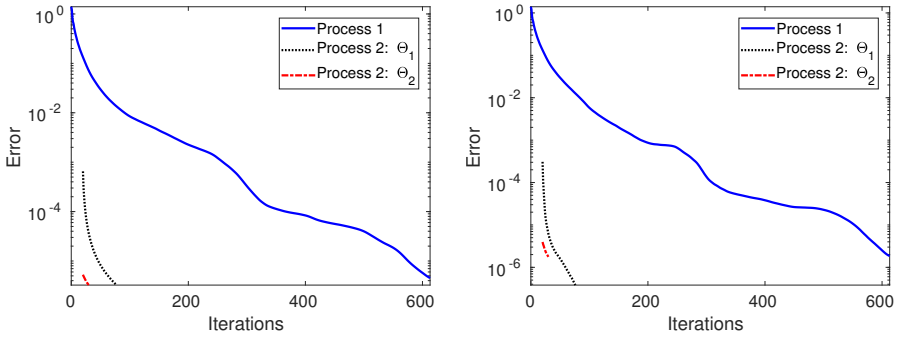
error and that of the algebraic error on $L$-marked elements for both processes but more markedly for the process 2. Now, when we increase the Dörfler rate to a value $\Theta_2$ close to one to cover almost all the high errors' region, we notice the rapid decrease of the global energy norm of the error and that of the algebraic error on $L$-marked elements with process 2 (black curve). The convergence is faster and there is a 3x and a major 14x speedups in terms of iterations on meshes $\mathcal{M}^{(1)}$ and $\mathcal{M}^{(2)}$ respectively (see Table 2).

We consider now the second test case. Figures 16 and 18 display the initial error distribution over meshes $\mathcal{M}^{(1)}$ and $\mathcal{M}^{(2)}$. Figures 17 and 19 depict the evolution of the global energy norm and the $L$-norm of the error for $\mathcal{M}^{(1)}$ and $\mathcal{M}^{(2)}$ with two different values of the parameter $\Theta$ in the Dörfler criterion. In those latter figures, we point out that the error reduction in process 2 with $\Theta_1$ is better than in process 1. It becomes even more efficient with the larger $\Theta_2$ as it yields a convergence after 65 or 11 iterations only according to the mesh resolution (see Table 2). That represents an important 5x (resp. 56x) speedup




(a) Algebraic a posteriori error estimates after 20 iterations (b) Algebraic errors after 20 iterations

**Fig. 18** Initial distribution and a posteriori estimation of algebraic error for test case n°2 on mesh $\mathcal{M}^{(2)}$




(a) Energy norm of the algebraic error (b) $L$-norm of the algebraic error

**Fig. 19** Error evolution for test case n°2 on mesh $\mathcal{M}^{(2)}$ using a posteriori error estimates

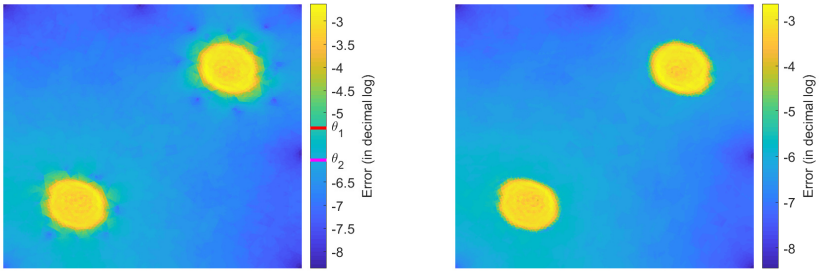
over PCG. These results show the efficiency of the adaptive method even for the case when the errors are localized in separate subdomains.

## 5.5 Numerical results for the second test case with a posteriori error estimates and $P_2$ polynomial approximation

We evaluate now the efficiency of the adaptive procedure for the second test case (2 peaks) of Section 5.2 with $P_2$ polynomial approximation, using a posteriori algebraic error estimates, without evaluating the exact algebraic error. The mesh configuration is the same as in Section 5.4. The initial distribution and a posteriori estimation of algebraic error (after $j_0 = 20$ iterations) over the domain $\Omega$ are shown in Figure 20 for the first mesh $\mathcal{M}^{(1)}$ and in Figure 22 for the second mesh $\mathcal{M}^{(2)}$. The global energy norm and the $L$-norm of the error during the iterations are shown in Figures 21 (for $\mathcal{M}^{(1)}$) and 23 (for $\mathcal{M}^{(2)}$) with two different values of the parameter $\Theta$ in the Dörfler criterion. The test

**Table 2** Test configuration and number of iterations for standard and adaptive processes applied to Poisson problems using local error estimates with different values of the Dörfler rate $\Theta$.

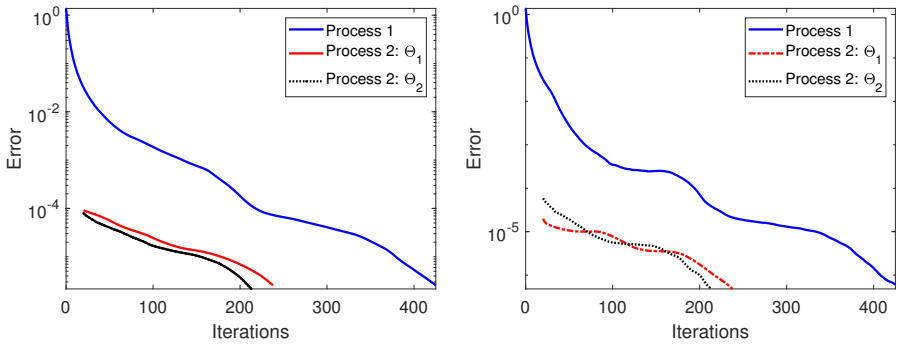
| Configuration | | | | Iterations | |
|---|---|---|---|---|---|
| Test case | Mesh | $\Theta$ (as %) | $n_L$ in % of $n$ | $it_{\text{st}}$ | $it_{\text{ada}}$ |
| 1 | $\mathcal{M}^{(1)}$ | 87.70 | 3.50 | 278 | 230 |
| | | 99.99 | 10.99 | 278 | 92 |
| 1 | $\mathcal{M}^{(2)}$ | 99 | 9.99 | 689 | 322 |
| | | 99.99 | 15 | 689 | 47 |
| 2 | $\mathcal{M}^{(1)}$ | 99.82 | 10 | 314 | 96 |
| | | 99.99 | 25.17 | 314 | 65 |
| 2 | $\mathcal{M}^{(2)}$ | 99 | 3.99 | 614 | 57 |
| | | 99.99 | 10 | 614 | 11 |




(a) Algebraic a posteriori error estimates after 20 iterations

(b) Algebraic errors after 20 iterations

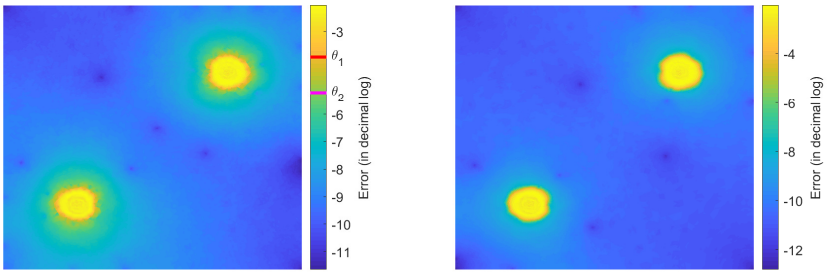
**Fig. 20** Initial distribution and a posteriori estimation of algebraic error for test case n°2 on mesh $\mathcal{M}^{(1)}$. $P_2$ polynomial approximation is used here.




(a) Energy norm of the algebraic error

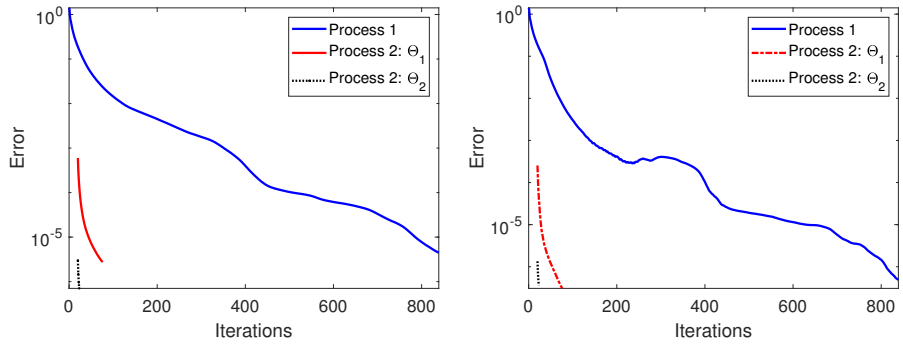
(b) $L$-norm of the algebraic error

**Fig. 21** Error evolution for test case n°2 on mesh $\mathcal{M}^{(1)}$ using a posteriori error estimates and $P_2$ polynomial approximation.




(a) Algebraic a posteriori error estimates after 20 iterations (b) Algebraic errors after 20 iterations

**Fig. 22** Initial distribution and a posteriori estimation of algebraic error for test case n°2 on mesh $\mathcal{M}^{(2)}$. $P_2$ polynomial approximation is used here.




(a) Energy norm of the algebraic error (b) $L$-norm of the algebraic error

**Fig. 23** Error evolution for test case n°2 on mesh $\mathcal{M}^{(2)}$ using a posteriori error estimates and $P_2$ polynomial approximation.

**Table 3** Test configuration and number of iterations for standard and adaptive processes applied to test case n°2 using local error estimates with different values of the Dörfler rate $\Theta$. $P_2$ polynomial approximation is used here.

| **Configuration** | | | | **Iterations** | |
|---|---|---|---|---|---|
| Test case | Mesh | $\Theta$ (as %) | $n_L$ in % of $n$ | $it_{\text{st}}$ | $it_{\text{ada}}$ |
| 2 | $\mathcal{M}^{(1)}$ | 99 | 10 | 425 | 218 |
| | | 99.99 | 25.17 | 425 | 193 |
| 2 | $\mathcal{M}^{(2)}$ | 99 | 2.50 | 840 | 56 |
| | | 99.99 | 3.99 | 840 | 3 |

configurations and results are detailed in Table 3. We notice that the adaptive procedure is still effective, as the number of iterations is reduced by more than half for the mesh $\mathcal{M}^{(1)}$, and by ten times for the mesh $\mathcal{M}^{(2)}$.

### 5.6 Concluding remarks on the numerical experiments

In the light of of the numerical results presented above, we can draw the following observations:

- The solve process proposed in this paper is adaptive as it adjusts to the considered problem thanks to the information stemming from the a posteriori estimation of the algebraic error. It seems to perform well provided that $\Omega_1$ is appropriately built, i.e. we mark and gather in $\Omega_1$ enough (or all ideally) elements that really reflect the regions of the domain where the resolution of the system is more delicate and requires some special local treatment.
- The adaptive process is better suited for test cases where there is a considerable discrepancy in errors and a concentration in space that is conducive to adaptivity (like (34) and (35)). In such cases, the elements' marking is facilitated by the fact that the high algebraic errors seem to properly represent the potential regions that are of interest to us in order to build $\Omega_1$. Herein, one can see an analogy with adaptive refinement in Finite Element Methods.
- In addition, this procedure seems to perform poorly when the algebraic errors are widely spread in the domain. In such cases, it becomes costly to include all elements with significant errors inside $\Omega_1$ since we will have to factorize the $\mathbf{A}_L$ submatrix. And when we fill that subdomain with only a part of those elements, the procedure yields unsatisfactory results as we can see with test case n°4. When we include all the region of large errors inside $\Omega_1$, the adaptive procedure performs well as we observed in test cases n°1, n°2 and n°4.
- Another issue relates to the spread of the error through the whole domain. We notice that the adaptive procedure performs better when the area with large errors is contiguous. We can clearly see that by comparing test cases n°3 and n°4.

## 6 Conclusions

In this paper, we have presented a new adaptive preconditioner for iterative solution of sparse linear systems arising from PDE problems that is used in combination with a specific initial guess and based on the estimated local distribution of the algebraic error. The proposed adaptive procedure aims at efficiently reducing the algebraic error norm by targeting the regions where the algebraic error is high. As shown in numerical experiments, in the case of an important discrepancy in the algebraic error, when it is dominating

in certain parts of the domain, notable speedups can be achieved with the proposed procedure: with a proper treatment on high-error regions, the number of iterations can be significantly diminished.
We are aware that there is a lot of more work to be done in order to derive a robust practically applicable and efficient procedure. Nevertheless, the present study has confirmed that the concept of adaptivity based on the (local) distribution of the algebraic error is worthy considering. A follow-up direction could be to investigate a more general algorithm not necessarily requiring that the L-subdomain solve is carried out exactly (as suggested e.g. in [12]).

## References


1. Arioli, M., Liesen, J., Miedlar, A., Strakoš, Z.: Interplay between discretization and algebraic computation in adaptive numerical solutionof elliptic pde problems. GAMM-Mitteilungen **36**(1), 102–129 (2013). DOI 10.1002/gamm.201310006
2. Babuška, I., Rheinboldt, W.C.: Error estimates for adaptive finite element computations. SIAM J. Numer. Anal. **15**(4), 736–754 (1978)
3. Becker, R., Johnson, C., Rannacher, R.: Adaptive error control for multigrid finite element methods. Computing **55**(4), 271–288 (1995). DOI 10.1007/BF02238483
4. Carstensen, C.: A posteriori error estimate for the mixed finite element method. Math. Comp. **66**(218), 465–476 (1997). DOI 10.1090/S0025-5718-97-00837-5
5. Chan, T.F., Mathew, T.P.: Domain decomposition algorithms. In: Acta Numerica 1994, pp. 61–143. Cambridge University Press (1994)
6. Di Pietro, D.A., Vohralík, M., Yousef, S.: Adaptive regularization, linearization, and discretization and a posteriori error control for the two-phase Stefan problem. Math. Comp. **84**(291), 153–186 (2015). DOI 10.1090/S0025-5718-2014-02854-8
7. Dörfler, W.: A convergent adaptive algorithm for Poisson's equation. SIAM J. Numer. Anal. **33**(3), 1106–1124 (1996). DOI 10.1137/0733054
8. Ern, A., Vohralík, M.: Adaptive inexact Newton methods with a posteriori stopping criteria for nonlinear diffusion PDEs. SIAM J. Sci. Comput. **35**(4), A1761–A1791 (2013). DOI 10.1137/120896918
9. Golub, G.H., Strakoš, Z.: Estimates in quadratic formulas. Numerical Algorithms **8**(2), 241–268 (1994). DOI 10.1007/BF02142693
10. Jiránek, P., Strakoš, Z., Vohralík, M.: A posteriori error estimates including algebraic error and stopping criteria for iterative solvers. SIAM Journal on Scientific Computing **32**(3), 1567–1590 (2010). DOI 10.1137/08073706X
11. Jolivet, P., Dolean, V., Hecht, F., Nataf, F., Prud'homme, C., Spillane, N.: High performance domain decomposition methods on massively parallel architectures with freefem++. Journal of Numerical Mathematics **20**, 287302 (2013)
12. Keyes, D.E., Gropp, W.D.: A comparison of domain decomposition techniques for elliptic partial differential equations and their parallel implementation. SIAM Journal on Scientific and Statistical Computing **8**(2), s166–s202 (1987). DOI 10.1137/0908020
13. Li, Z., Saad, Y., Sosonkina, M.: parms: a parallel version of the algebraic recursive multilevel solver. Numerical Linear Algebra with Applications **10**(5-6), 485–509 (2003). DOI 10.1002/nla.325
14. Mandel, J.: On block diagonal and schur complement preconditioning. Numerische Mathematik **58**(1), 79–93 (1990)
15. Papež, J., Liesen, J., Strakoš, Z.: Distribution of the discretization and algebraic error in numerical solution of partial differential equations. Linear Algebra and its Applications **449**, 89 – 114 (2014). DOI 10.1016/j.laa.2014.02.009
16. Papež, J., Rüde, U., Vohralík, M., Wohlmuth, B.: Sharp algebraic and total a posteriori error bounds for $h$ and $p$ finite elements via a multilevel approach (2017). URL `https://hal.inria.fr/hal-01662944`. HAL-preprint

17. Papež, J., Strakoš, Z., Vohralík, M.: Estimating and localizing the algebraic and total numerical errors using flux reconstructions. Numerische Mathematik **138**(3), 681–721 (2018). DOI 10.1007/s00211-017-0915-5
18. Saad, Y.: Iterative Methods for Sparse Linear Systems, second edn. SIAM (2000)
19. Saad, Y., Sosonkina, M.: Distributed schur complement techniques for general sparse linear systems. SIAM Journal on Scientific Computing **21**(4), 1337–1356 (1999). DOI 10.1137/S1064827597328996
20. Verfürth, R.: A review of a posteriori error estimation and adaptive mesh-refinement techniques. Teubner-Wiley, Stuttgart (1996)
21. Vohralík, M., Yousef, S.: A simple a posteriori estimate on general polytopal meshes with applications to complex porous media flows. Computer Methods in Applied Mechanics and Engineering **331**, 728 – 760 (2018). DOI 10.1016/j.cma.2017.11.027

## Appendix

A posteriori error estimates

In this section we describe briefly one of the ways how to estimate the local distribution of the algebraic error (6) using flux reconstructions following [10, 8,17] and references therein. We start by introducing the basic techniques of these a posteriori error estimates, then we detail how to get a sharp computable upper bound of the algebraic error. Note that this section recalls existing techniques and results and we only adapt the a posteriori error estimate of [17] to our chosen model problem.

*Basic a posteriori error estimates*

Assuming that a Galerkin solution $u_h$ is available, we start by bounding the energy norm of the error $\underline{u} - \underline{u}_h$ represented as

$$\|\underline{\mathbf{K}}^{\frac{1}{2}} \nabla(\underline{u} - \underline{u}_h)\|_{\mathrm{L}^2(\Omega)} = \sup_{\underline{v} \in V, \|\nabla \underline{v}\| = 1} (\underline{\mathbf{K}}^{\frac{1}{2}} \nabla(\underline{u} - \underline{u}_h), \underline{\mathbf{K}}^{\frac{1}{2}} \nabla \underline{v}). \tag{39}$$

Note that following (2), then

$$\|\underline{\mathbf{K}}^{\frac{1}{2}} \nabla(\underline{u} - \underline{u}_h)\|_{\mathrm{L}^2(\Omega)} = \sup_{\underline{v} \in V, \|\nabla \underline{v}\| = 1} \{(\underline{f}, \underline{v}) - (\underline{\mathbf{K}}^{\frac{1}{2}} \nabla \underline{u}_h, \underline{\mathbf{K}}^{\frac{1}{2}} \nabla \underline{v})\}. \tag{40}$$

The key ingredient of our estimate is a reconstructed flux $\boldsymbol{\theta}_h$ which is a piecewise polynomial function in the Raviart–Thomas–Nédélec subspace $\mathbf{RTN}(\mathcal{T}_h)$ of the infinite-dimensional space $\mathbf{H}(\mathrm{div}, \Omega)$ which is reconstructed in order to mimic the continuous flux $\boldsymbol{\theta} := -\underline{\mathbf{K}} \nabla \underline{u}$. In other words, $\boldsymbol{\theta}_h$ is reconstructed to satisfy

$$\nabla \cdot \boldsymbol{\theta}_h = \underline{f}. \tag{41}$$

Recall from Section 2 that $\underline{f}$ is assumed to be piecewise constant with respect to the mesh $\mathcal{T}_h$. Now, we use the Green and the Cauchy–Schwarz inequality

together with (40) and we follow [17, Section 4.1], to write

$$\begin{aligned}\|\underline{\mathbf{K}}^{\frac{1}{2}}\nabla(\underline{u}-\underline{u}_h)\|_{\mathrm{L}^2(\Omega)} &= \inf_{\substack{\boldsymbol{\sigma}\in\mathbf{H}(\mathrm{div},\Omega)\\ \nabla\cdot\boldsymbol{\sigma}=\underline{f}}} \sup_{\substack{\underline{v}\in V\\ \|\nabla\underline{v}\|=1}} \{(\underline{f}-\nabla\cdot\boldsymbol{\sigma},\underline{v}) \\ &\qquad -(\underline{\mathbf{K}}^{\frac{1}{2}}\nabla\underline{u}_h+\underline{\mathbf{K}}^{-\frac{1}{2}}\boldsymbol{\sigma},\underline{\mathbf{K}}^{\frac{1}{2}}\nabla\underline{v})\} \\ &= \inf_{\substack{\boldsymbol{\sigma}\in\mathbf{H}(\mathrm{div},\Omega)\\ \nabla\cdot\boldsymbol{\sigma}=\underline{f}}} \sup_{\substack{\underline{v}\in V\\ \|\nabla\underline{v}\|=1}} \{-(\underline{\mathbf{K}}^{\frac{1}{2}}\nabla\underline{u}_h+\underline{\mathbf{K}}^{-\frac{1}{2}}\boldsymbol{\sigma},\underline{\mathbf{K}}^{\frac{1}{2}}\nabla\underline{v})\} \\ &= \inf_{\substack{\boldsymbol{\sigma}\in\mathbf{H}(\mathrm{div},\Omega)\\ \nabla\cdot\boldsymbol{\sigma}=\underline{f}}} \|\underline{\mathbf{K}}^{\frac{1}{2}}\nabla\underline{u}_h+\underline{\mathbf{K}}^{-\frac{1}{2}}\boldsymbol{\sigma}\|_{\mathrm{L}^2(\Omega)} \\ &\leq \|\underline{\mathbf{K}}^{\frac{1}{2}}\nabla\underline{u}_h+\underline{\mathbf{K}}^{-\frac{1}{2}}\boldsymbol{\theta}_h\|_{\mathrm{L}^2(\Omega)}.\end{aligned}$$

This gives a guaranteed upper bound on the discretization error. Note that the obtained estimate relies only on the weak formulation and the reconstructed flux $\boldsymbol{\theta}_h$. Finally, it is important to mention that the needed reconstructed flux $\boldsymbol{\theta}_h$ can be easily reconstructed for various discretization schemes like finite elements, nonconforming finite elements, discontinuous Galerkin, finite volumes, and mixed finite elements, see [8] for more details.

*Upper bound on the algebraic error*

In this section we suppose that we use an iterative solver to obtain an approximate solution $\underline{u}_h^{(i)}$ of (3) after running $i$ iterations. In order to estimate the algebraic error we first introduce a representation of the algebraic residual vector which will be a function $\underline{s}_h^{(i)} \in \mathrm{L}^2(\Omega)$ satisfying

$$(\underline{s}_h^{(i)},\varphi_j) = \mathbf{r}_j^{(i)}, \qquad 1\leq j\leq n. \tag{42}$$

Details about the reconstruction of $\underline{s}_h^{(i)}$ with two differents examples can be found in [17, Section 5.1]. Note that from (42) and the definition of the algebraic residual vector $\mathbf{r}^{(i)}$ one can write

$$(\underline{s}_h^{(i)},\varphi_j) = (\underline{f},\varphi_j) - (\underline{\mathbf{K}}^{\frac{1}{2}}\nabla\underline{u}_h^{(i)},\underline{\mathbf{K}}^{\frac{1}{2}}\nabla\varphi_j), \qquad 1\leq j\leq n. \tag{43}$$

Consequently,

$$(\underline{s}_h^{(i)},\underline{v}_h) = (\underline{f},\underline{v}_h) - (\underline{\mathbf{K}}^{\frac{1}{2}}\nabla\underline{u}_h^{(i)},\underline{\mathbf{K}}^{\frac{1}{2}}\nabla\underline{v}_h). \tag{44}$$

Using (3), then (44) gives

$$(\underline{s}_h^{(i)},\underline{v}_h) = (\underline{\mathbf{K}}^{\frac{1}{2}}(\nabla\underline{u}_h-\nabla\underline{u}_h^{(i)}),\underline{\mathbf{K}}^{\frac{1}{2}}\nabla\underline{v}_h). \tag{45}$$

This representation of the algebraic residual vector plays a key role in the estimation of the algebraic error. Actually, by applying the Cauchy-Schwarz

inequality together with the Friedrichs inequality on (45), one gets

$$\begin{aligned}(\underline{\mathbf{K}}^{\frac{1}{2}}(\nabla\underline{u}_h - \nabla\underline{u}_h^{(i)}), \underline{\mathbf{K}}^{\frac{1}{2}}\nabla\underline{v}_h) &= (\underline{s}_h^{(i)}, \underline{v}_h)\\ &\leq \|\underline{s}_h^{(i)}\|_{\mathrm{L}^2(\Omega)}\|\underline{v}_h\|_{\mathrm{L}^2(\Omega)}\\ &\leq \|\underline{s}_h^{(i)}\|_{\mathrm{L}^2(\Omega)}\left(C_{\mathrm{F}}h_\Omega\|\nabla\underline{v}_h\|_{\mathrm{L}^2(\Omega)}\right)\\ &\leq \|\underline{s}_h^{(i)}\|_{\mathrm{L}^2(\Omega)}\left(C_{\mathrm{F}}h_\Omega\lambda_{\underline{\mathbf{K}}}^{-\frac{1}{2}}\|\underline{\mathbf{K}}^{\frac{1}{2}}\nabla\underline{v}_h\|_{\mathrm{L}^2(\Omega)}\right),\end{aligned}$$

where $0 < C_{\mathrm{F}} \leq 1$ is the constant from the Friedrichs inequality, $h_\Omega$ the diameter of the domain $\Omega$, and $\lambda_{\underline{\mathbf{K}}}$ the smallest eigenvalue of $\underline{\mathbf{K}}$. First computable upper bound is then obtained for the algebraic error as

$$\|\underline{\mathbf{K}}^{\frac{1}{2}}\nabla(\underline{u}_h - \underline{u}_h^{(i)})\|_{\mathrm{L}^2(\Omega)} \leq C_{\mathrm{F}}h_\Omega\lambda_{\underline{\mathbf{K}}}^{-\frac{1}{2}}\|\underline{s}_h^{(i)}\|_{\mathrm{L}^2(\Omega)}. \tag{46}$$

However, this upper bound yields often a significant overestimation; see, e.g., [17, Sections 3.1 and 4.2]. An improvement of the upper bound (46) can be obtained by using flux reconstruction techniques and additional algebraic iterations. Following Section 6 and [17, Section 5.3], we consider a reconstructed flux $\boldsymbol{\theta}_h^{(i)} \in \mathbf{RTN}(\mathcal{T}_h)$ satisfying $\nabla\cdot\boldsymbol{\theta}_h^{(i)} = \underline{f} - \underline{s}_h^{(i)}$. Then, after $\nu > 0$ additional iterations we similarly construct from the algebraic residual vector $r^{(i+\nu)}$ a representation $\underline{s}_h^{(i+\nu)}$, and another reconstructed flux $\boldsymbol{\theta}_h^{(i+\nu)} \in \mathbf{RTN}(\mathcal{T}_h)$ satisfying $\nabla\cdot\boldsymbol{\theta}_h^{(i+\nu)} = \underline{f} - \underline{s}_h^{(i+\nu)}$. With these different reconstructions we have

$$\underline{s}_h^{(i)} = \underline{s}_h^{(i+\nu)} + \nabla\cdot\boldsymbol{\theta}_h^{(i+\nu)} - \nabla\cdot\boldsymbol{\theta}_h^{(i)},$$

and therefore (45) gives

$$(\underline{\mathbf{K}}^{\frac{1}{2}}(\nabla\underline{u}_h - \nabla\underline{u}_h^{(i)}), \underline{\mathbf{K}}^{\frac{1}{2}}\nabla\underline{v}_h) = (\underline{\mathbf{K}}^{-\frac{1}{2}}(\boldsymbol{\theta}_h^{(i+\nu)} - \boldsymbol{\theta}_h^{(i)}), \underline{\mathbf{K}}^{\frac{1}{2}}\nabla\underline{v}_h) + (\underline{s}_h^{(i+\nu)}, \underline{v}_h).$$

Consequently,

$$\|\underline{\mathbf{K}}^{\frac{1}{2}}\nabla(\underline{u}_h - \underline{u}_h^{(i)})\|_{\mathrm{L}^2(\Omega)} \leq \|\underline{\mathbf{K}}^{-\frac{1}{2}}(\boldsymbol{\theta}_h^{(i+\nu)} - \boldsymbol{\theta}_h^{(i)})\|_{\mathrm{L}^2(\Omega)} + C_{\mathrm{F}}h_\Omega\lambda_{\underline{\mathbf{K}}}^{-\frac{1}{2}}\|\underline{s}_h^{(i+\nu)}\|_{\mathrm{L}^2(\Omega)} \tag{47}$$

The idea of using additional algebraic iterations is very useful in practice [9]. In fact, for a sufficiently large value of $\nu$ one can assume that there exists $\gamma > 0$ such that

$$C_{\mathrm{F}}h_\Omega\lambda_{\underline{\mathbf{K}}}^{-\frac{1}{2}}\|\underline{s}_h^{(i+\nu)}\|_{\mathrm{L}^2(\Omega)} \leq \gamma\|\underline{\mathbf{K}}^{-\frac{1}{2}}(\boldsymbol{\theta}_h^{(i+\nu)} - \boldsymbol{\theta}_h^{(i)})\|_{\mathrm{L}^2(\Omega)},$$

so that

$$\|\underline{\mathbf{K}}^{\frac{1}{2}}\nabla(\underline{u}_h - \underline{u}_h^{(i)})\|_{\mathrm{L}^2(\Omega)} \leq (1+\gamma)\|\underline{\mathbf{K}}^{-\frac{1}{2}}(\boldsymbol{\theta}_h^{(i+\nu)} - \boldsymbol{\theta}_h^{(i)})\|_{\mathrm{L}^2(\Omega)},$$

which gives an upper bound easily computed and cheaply evaluated in practice even for complex problems, see [21] for details.

*Remark 5 (Local indicators for the algebraic error)* In order to estimate the local distribution of the algebraic error (6) using flux reconstructions, one can use the local indicators $\eta^{(i)}_{\mathrm{alg},K} := \|\underline{\mathbf{K}}^{-\frac{1}{2}}(\boldsymbol{\theta}_h^{(i+\nu)} - \boldsymbol{\theta}_h^{(i)})\|_{\mathrm{L}^2(K)} + C_{\mathrm{F}} h_\Omega \lambda_{\underline{\mathbf{K}}}^{-\frac{1}{2}} \|\underline{s}_h^{(i+\nu)}\|_{\mathrm{L}^2(K)}$. Relying on the previous discussion, in practice with a sufficiently large $\nu$ one can use the local algebraic indicator $\|\underline{\mathbf{K}}^{-\frac{1}{2}}(\boldsymbol{\theta}_h^{(i+\nu)} - \boldsymbol{\theta}_h^{(i)})\|_{\mathrm{L}^2(K)}$ which can be the ingredient of our adaptive procedure.

*Remark 6 (A posteriori error estimates for the total error)* A computable upper bound can be obtained on the energy norm of the total error $\underline{u} - \underline{u}_h^{(i)}$ following the same ideas of Section 6 and Section 6:

$$\|\underline{\mathbf{K}}^{\frac{1}{2}} \nabla(\underline{u} - \underline{u}_h^{(i)})\|_{\mathrm{L}^2(\Omega)} \leq \eta^{(i)}_{\mathrm{dis}} + \eta^{(i)}_{\mathrm{alg}}$$

with

$$\eta^{(i)}_{\mathrm{dis}} := \|\underline{\mathbf{K}}^{\frac{1}{2}} \nabla \underline{u}_h^{(i)} + \underline{\mathbf{K}}^{-\frac{1}{2}} \boldsymbol{\theta}_h^{(i)}\|_{\mathrm{L}^2(\Omega)}$$

and

$$\eta^{(i)}_{\mathrm{alg}} := \|\underline{\mathbf{K}}^{-\frac{1}{2}}(\boldsymbol{\theta}_h^{(i+\nu)} - \boldsymbol{\theta}_h^{(i)})\|_{\mathrm{L}^2(\Omega)} + C_{\mathrm{F}} h_\Omega \lambda_{\underline{\mathbf{K}}}^{-\frac{1}{2}} \|\underline{s}_h^{(i+\nu)}\|_{\mathrm{L}^2(\Omega)},$$

see [17] for the full demonstration.

*Remark 7 (A posteriori error estimates in the multilevel setting)* There is also another way how to construct upper bounds for the total and algebraic errors without the need of running additional iterations of the algebraic solver. The construction assumes the existence of the hierarchy of meshes, with a global solve on the coarsest mesh; see [16] for more details.